RAJIV GANDHI INSTITUTE OF TECHNOLOGY, KOTTAYAM

(Govt. Engineering College Kottayam)

DEPARTMENT OF COMPUTER APPLICATIONS

Since its inception in the year 2001, the department has been a fertile soil for budding professionals of Computer Applications. Armed with a bunch of experienced faculty and expert technical staff, it is well prepared to transform the students to meet the requirements of the information technology industry.

The students have already developed the campus Management software, 'RITSoft' and various campus related softwares. They have been actively involved in developing software systems for a lot of Govt. agencies.

Lab facilities

- Software Lab
- Hardware Lab

SALIENT FEATURES OF COURSE

- 2 year course with 60 seats
- Affiliated to KTU
- Well equipped laboratories
- Focus on practical knowledge

Achievements!!!

- Secured Rank 1 and 3 in KTU Exam 2021.
- Student involvment in Socially relevant projects:

  * Software for LAC Election 2021 for District Election Officer

  * Front office management system for Taluk office, Kottayam

  * Complaint redressal campaign of District Collector,Kottayam

Industry tie-ups:

IBS
TCS
Infosys
Wipro etc...

CONTACT
0481 2507763

Velloor PO, Pampady, Kottayam
www.rit.ac.in

Mail ID:
hodmca@rit.ac.in

SEMESTER 1

SEMESTER – I

20MCA101	**MATHEMATICAL FOUNDATIONS FOR COMPUTING**	**CATEGORY**	**L**	**T**	**P**	**CREDIT**
		GENERAL	**3**	**1**	**0**	**4**

Preamble: This course introduces students to some basic mathematical ideas and tools which are at the core of MCA course. It introduces the concepts of graph theory, set theory and statistics.

Prerequisite: A basic course in set theory and statistics.

Course Outcomes: After the completion of the course the student will be able to

CO 1	Understand mathematical reasoning in order to read, comprehend and construct mathematical arguments
CO 2	Count or enumerate objects and solve counting problems and analyze algorithms
CO 3	Solve problems in almost every conceivable discipline using graph models
CO 4	Solve the linear system of equations and Calculate the eigen values and eigen vectors of matrices.
CO 5	Apply the principles of correlation and regression in practical problems.

Mapping of course outcomes with program outcomes

	PO 1	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO 10**	**PO 11**	**PO 12**
CO 1	3	3	3	3			3					
CO 2	3	3	3	3			3					
CO 3	3	3	3	3			3					
CO 4	3	3	3	3			3					
CO 5	3	3	3	3			3					

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination
	1	2	
Remember(K1)	10	10	10
Understand(K2)	20	20	20
Apply(K3)	20	20	30
Analyse(K4)			
Evaluate(K5)			
Create(K6)			

Mark distribution

Total Marks	CIE	ESE	ESE Duration
100	40	60	3 hours

Continuous Internal Evaluation Pattern:

Attendance	: 8 marks
Continuous Assessment Test (2 numbers)	: 20 marks
Assignment/Quiz/Course project	: 12 marks

End Semester Examination Pattern: There will be two parts; Part A and Part B. Part A contains 10 compulsory short answer questions, 2 from each module. Each question caries 3 marks. Part B contains 2 questions from each module of which student should answer any one.Each question can have maximum 2 sub-divisions and carry 6 marks.

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Define equivalence relation with suitable example. (K1)

2. Write Warshall's algorithm. Use to find the transitive closure of the relation

   {(1,3), (3,2), (2,4), (3,1), (4,1) on (1,2,3,4)} (K2)

3. Let f,g:R□R be defined by f(x)= x+1, g(x)= $2x^2+3$, find fog and gof. Is fog = gof ? (K3)

Course Outcome 2 (CO2)

1. Solve the linear Diophantine equation 24x+138y = 18 (K5)
2. Find the GCD (12378,3054) (K3)
3. Solve a_{n+2}- 4a_{n+1}+3a_n= -200,n □0 given that a_o= 3000, a_1=3300 (K5)

Course Outcome 3(CO3):

1. Define Hamilton cycle and Euler circuit with example. (K1)

2. Show that $K_{3,3}$ is non-planar. Define planar graph. State Kuratowski's theorem. (K4)

3. Prove that a connected graph G is an Euler graph if all vertices of G are of even degree. (K4)

Course Outcome 4 (CO4):

1. Find the rank of the matrix $\begin{bmatrix} 0 & 3 & 4 \\ -3 & 0 & -5 \\ -4 & 5 & 0 \end{bmatrix}$ (K3)
2. Find the Eigen values and Eigen vectors of $\begin{bmatrix} 4 & 2 & -2 \\ 2 & 5 & 0 \\ -2 & 0 & 3 \end{bmatrix}$ (K3)
3. Find out what type of conic sections the quadratic form $Q = 17x_1^2 - 30x_1x_2 + 17x_2^2 = 128$ represents and transform it into principal axes form (K3)

Course Outcome 5 (CO5):

1. State the principle of least squares. (K1)
2. Fit a parabola by the method of least squares, to the following data. (K3)

x:	1	2	3	4	5
y:	5	12	26	60	97

3. Compute the correlation coefficient from the following data. (K3)

x:	77	54	27	52	14	35	90	25	96	60
y:	35	58	60	40	50	40	35	56	34	42

Syllabus

Module 1

Sets, Set Operations, Relations, Classification of relations, Equivalence Relations, Closures of Relations, Matrix Representation of Relations, Partial Ordering, n-ary Relations, Functions.

Module 2

Division Algorithm, GCD, Primes, Euclidean Algorithm, Congruences, Properties of Congruences, Solutions of Linear Congruences.

First Order Linear Recurrence Relation, Second Order Linear Homogeneous Recurrence Relations with Constant coefficients, Non Homogeneous Recurrence Relation.

Module 3

Graphs and Graph Models, Graph Terminology and Special Types of Graphs, Representing Graphs and Graph Isomorphism, Connectivity, Euler and Hamilton Paths, Shortest-Path Problems, Planar Graphs, Directed Graph, Multigraph, Connected graph, Euler circuit and trail, Planar and Non-planar Graphs.

Module 4

Linear system of equations, coefficient matrix, augmented matrix, Gauss elimination method and back substitution, elementary row operations, row equivalent systems, Gauss elimination- three possible cases, Row Echelon form and information from it, Linear independence- rank of a matrix. Solution of linear system, fundamental theorem of non- homogeneous linear system (without proof). Homogeneous linear system (theory only), Matrix eigen value problem- determination of eigen values and eigen vectors, Basis of eigen vectors- diagonalization of matrix- Quadratic form- principle axis theorem (without proof).

Module 5

Bivariate data – Scatter Diagram – Interpretation of the nature and degree of relation using scattered diagram - Curve fitting – Principle of least squares – fitting a straight line – fitting a parabola – linear correlation and regression – Karl’s Pearson’s Coefficient of Correlation – Spearman’s rank correlation coefficient (problems based on the formula).

Text Books

1. David M. Burton, “Elementary Number Theory”, McGraw-Hill, 7th Edition (2012).

2. Ralph P Grimaldi, “Discrete and Computational Mathematics: An applied introduction”, Pearson Education, 5th Edition, (2007).

3. Erwin Kreyszig, “Advanced Engineering Mathematics”, 10th ed., Wiley.
4. Gupta S.C and Kapoor V .K, “Fundamentals of Mathematical Statistics”, Sultan Chand and Sons 11th edition.

Reference Books

1. C. Liu, “Elements of Discrete Mathematics: A Computer Oriented Approach”, McGraw-Hill, 4th Edition (2012).
2. Jean-Paul Tremblay ,“Discrete Mathematical Structures with applications to Computer science”, ”, McGraw-Hill, 1st Edition (2001).
3. Kenneth H. Rosen , ”Discrete mathematics and its applications”, McGraw-Hill, (7th Edition), (Smartbook available).
4. Marty Lewinter, Jeanine Meyer, “Elementary Number Theory with Programming”, Wiley- Blackwell (2015).
5. David S. Moore and George P. McCabe, “Introduction to practice of statistics”, W.H. Freeman & Company, 5th Edition (2005).
6. Douglas C. Montgomery and George C. Runger, “Applied Statistics and Probability for Engineers”, Wiley India, 5th Edition (2012).
7. Veerarajan T, “Probability and Random Process”, 3rd Edition, Tata McGraw-Hill (2002)
8. G. Jay Kerns, “Introduction to Probability and Statistics Using R”, Chapman & Hall (2010).
9. B.S Grewal. Higher Engineering Mathematics, Khanna Publishers, New Delhi.

Web Resources

1. Probability and statistics EBook
   http://wiki.stat.ucla.edu/socr/index.php/EBook
2. https://www.openintro.org/stat/textbook.php
3. http://www.math.uah.edu/stat/index.html
4. Statistics Online Computational Resource
   http://www.socr.ucla.edu/

Course Contents and Lecture Schedule

Topic	No. of lectures
Module 1	9 hrs.
Sets, Set Operations	2
Relations, Classification of relations, Equivalence Relations	2
Closures of Relations, Matrix Representation of Relations, Partial Ordering, n-ary Relations	3
Functions	2

Module 2	9 hrs.
Division Algorithm, GCD, Primes, Euclidean Algorithm	2
Congruences, Properties of Congruences, Solutions of Linear Congruences	2
First Order Linear Recurrence Relation	1
Second Order Linear homogeneous Recurrence Relations with Constant coefficients	2
Non Homogeneous Recurrence Relation	2
Module 3	8 hrs.
Graphs and Graph Models, Graph Terminology and Special Types of Graphs	1
Representing Graphs and Graph Isomorphism, Connectivity	2
Euler and Hamilton Paths, Shortest-Path Problems, Planar Graphs	2
Directed Graph, Multigraph, Connected graph	1
Euler circuit and trail, Planar and Non-Planar Graphs	2
Module 4	11 hrs.
Linear system of equations, coefficient matrix, augmented matrix, Gauss elimination method and back substitution, elementary row operations, row equivalent systems	2
Gauss elimination- three possible cases, Row Echelon form and information from it	2
Linear independence- rank of a matrix. Solution of linear system, fundamental theorem of non- homogeneous linear system (without proof). Homogeneous linear system (theory only), fundamental theorem of non- homogeneous linear system (without proof). Homogeneous linear system (theory only)	3
Matrix eigen value problem- determination of eigen values and eigen vectors, Basis of eigen vectors	2
diagonalization of matrix, Quadratic form-principle axis theorem (without proof).	2
Module 5	8 hrs.
Bivariate data – Scatter Diagram – Interpretation of the nature and degree of relation using scattered diagram	2
Curve fitting – Principle of least squares – fitting a straight line – fitting a parabola	2
linear correlation and regression – Karl's Pearson's Coefficient of Correlation	2
Spearman's rank correlation coefficient	2

20MCA103	DIGITAL FUNDAMENTALS & COMPUTER ARCHITECTURE	CATEGORY	L	T	P	CREDIT
		GENERAL	3	1	0	4

Preamble:

The primary aim of this course is to understand the fundamentals behind the digital logic design and gain the experience to design digital circuits and systems. Students should also acquire some understanding and appreciation of a computer system's functional components, their characteristics, performance and interactions. They need to understand the computer architecture in order to make best use of the software tools and computer languages they use to create programs.

Prerequisite: NIL

Course Outcomes: After the completion of the course the student will be able to

CO 1	Apply the basics of digital electronics to design and realize simple combinational logic circuits
CO 2	Apply the digital electronics principles to design sequential logic circuits.
CO 3	Understand the different design features of computer architecture, Five key components of a computer, processor and memory making technologies, addressing modes & instruction formats.
CO 4	Understand Processor logic design conventions and data path, pipelining and hazards, I/O organization, Interrupts and direct memory access
CO 5	Understand and different types of memories - RAM, ROM, Cache memory, virtual memory etc. Apply the different memory design techniques.
CO 6	Understand the concept of single board computers like Arduino, Raspberry Pi etc. and apply the same in practical applications.

Mapping of course outcomes with program outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	3	3	2	1	-	-	1	-	-	-	-	-
CO 2	3	3	2	1	-	-	1	-	-	-	-	-
CO 3	1	1	-	1	-	-	1	-	-	-	-	-
CO 4	1	1	-	-	-	-	1	-	-	-	-	-
CO 5	2	2	1	1	-	-	1	-	-	-	-	-
CO 6	1	1	2	-	2	-	2	2	2	-	2	2

Assessment Pattern

Bloom's Category	**Continuous Assessment Tests**		**End Semester Examination**
	1	**2**	
Remember	10	10	20
Understand	20	20	20
Apply	20	20	20
Analyse			
Evaluate			
Create			

Mark distribution

Total Marks	**CIE**	**ESE**	**ESE Duration**
100	40	60	3 hours

Continuous Internal Evaluation Pattern:

Attendance	: 8 marks
Continuous Assessment Test (2 numbers)	: 20 marks
Assignment/Quiz/Course project	: 12 marks

End Semester Examination Pattern: There will be two parts; Part A and Part B. Part A contain 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which student should answer any one. Each question can have maximum 2 sub-divisions and carry 6 marks.

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Minimize the Boolean Expression f(A,B,C) = Σm(1,3,5,6,7) using K-map.

2. Convert the decimal number 3.257 * 10^4 into single precision floating point binary representation

3. Express -31 in sign magnitude,1's complement and 2's complement notations

Course Outcome 2 (CO2)

1. Explain J-K flipflop with its truth table

2. Design an asynchronous decade counter.

3. Describe the working of a Parallel in Serial Out register.

Course Outcome 3 (CO3):

1. Describe the key components of a computer.

2. Define addressing mode. List 5 addressing modes with examples.

3. Differentiate between fixed length encoding and variable length encoding.

Course Outcome 4 (CO4):

1. Define pipeline, describe how pipeline improves the performance of the machine.

3. List different types of pipeline hazards with examples.

2. Explain how interrupts from multiple devices handled?

Course Outcome 5 (CO5):

1. Illustrate different cache mapping techniques with neat diagrams.

2. Discuss about Read Only Memories

3. Design 2M*32 memory module using 512K *8 static memory chips.

Course Outcome 6 (CO6):

No questions for university examination, for internal assessments practical assignment for configuring a PC / arduino or raspberry and programming assignments using HDL like Verilog or VHDL can be given.

Model Question paper

Reg No.:_______________		Name:________________________________ ___

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

FIRST SEMESTER M.C.A.DEGREE EXAMINATION, MODEL QUESTION PAPER

Course Code: 20MCA103

Course Name: DIGITAL FUNDAMENTALS & COMPUTER ARCHITECTURE

Max. Marks: 60		Duration: 3 Hours

PART A

		Answer all questions, each carries 3 marks.	Marks
1		Represent +45,-45 in 1’s complement and 2’s complement form.	(3)
2		Implement a full adder using 8:1 MUX	(3)
3		How could you convert RS flip flop to D flip flop?	(3)
4		What is meant by modulus of a counter? Realize a mod-8 synchronous counter.	(3)
5		Suppose we have two implementations of the same instruction set architecture. Computer A has a clock cycle time of 250 ps and a CPI of 2.0 for some program, and computer B has a clock cycle time of 500 ps and a CPI of 1.2 for the same program. Which computer is faster for this program and by how much?	(3)
6		Describe about little endian and big endian byte ordering.	(3)
7		Explain 4 stage pipelining with a diagram.	(3)
8		Differentiate between memory mapped I/O and Isolated I/O	(3)
9		What is static RAM ?	(3)
10		Define temporal locality and spatial locality.	(3)

PART B

Answer any one question from each module. Each question carries 6 marks.

Module I

11		Explain about single precision floating point representation with an example	(6)
		OR	
12		Minimize the Boolean expression f(A,B,C,D)=Σm(1,5,6,7,9,15)+d(2,3,11,13) using Karnaugh map and realize it using NAND gates.	(6)

Module II			
13		Demonstrate the working of a JK flip flop. How does it eliminate the invalid condition in SR flip flop? List out its applications.	(6)
OR			
14		Design a mod-12 asynchronous counter.	(6)
Module III			
15		Explain the five classic components of a computer with diagram.	(6)
OR			
16		Describe the code sequence of C=A+B in different types of instruction set architecture.	(6)
Module IV			
17		Draw a single datapath representation for memory instructions and R-type instructions	(6)
OR			
18		What is Direct Memory Access? Explain two types of bus arbitration schemes	(6)
Module V			
19		Elaborate the various cache memory mapping techniques with an example for each.	(6)
OR			
20		Explain the internal organization of memory chips and design a 1K*1 memory chip using decoder.	(6)

Syllabus

Module I (11 Hours)

Representation of signed numbers – 1's complement and 2's complement ,Logic gates – AND - OR – NOT - NAND- NOR - XOR , Boolean algebra - Basic laws and theorems , Boolean functions - truth table, Standard forms of Boolean Expressions – Sum of Products and Product of Sums - minimization of Boolean function using Karnaugh map method - Realization using logic gates, Floating point numbers
Combinational Circuits - Half adder - Full Adder- Decoder -Encoder- Multiplexer – Demultiplexer

Module II (10 Hours)

Sequential circuit - Clocking, Flip flops - SR – JK- D -T flip flops, Counters - Synchronous and asynchronous counters - UP/DOWN counters , Registers - Serial in serial out - Serial in parallel out - Parallel in serial out - Parallel in parallel out registers

A practical assignments may be given in configuring a PC / configuring arduino - Implementing simple programs for blinking an LED - Input from an external switch - fading an LED - serial monitor and debugging / installing & configuring Raspberry pi.

Module III (10 Hours)

Computer abstractions and technology - Introduction, Computer architecture -8 Design features, Application program - layers of abstraction, Five key components of a computer, Technologies for building processors and memory, Performance, Instruction set principles – Introduction, Classifying instruction set architectures, Memory addressing, Encoding an instruction set.

Module IV (9 Hours)

The Processor - Introduction, Logic design conventions, Building a datapath, A simple implementation scheme, An overview of pipelining - Pipelined datapath and control - Structural hazards - Data hazards - Control hazards
I/O organization - Accessing I/O devices, interrupts - handling multiple devices, Direct memory access
Programming assignments may be given in any HDL like Verilog or VHDL to create gate level/ Dataflow/Behavioural level models of gates, multiplexers, adders, flip-flops, registers etc. No detailed teaching of HDL is necessary. The students can be given a basic tutorial write up on gate level modelling.

Module V (8 Hours)

The Memory System – basic concepts, semiconductor RAM memories - organization – static and dynamic RAM, Structure of larger memories, semiconductor ROM memories, Speed, Size and cost ,Cache memory – mapping functions – replacement algorithms , Virtual memory – paging and segmentation.

Text Books

1. Floyd, "***Digital Fundamentals***", Pearson Education, 10th Edition (2011).(Module 1 & 2)
2. J. Hennessy and D. Patterson, "***Computer Organization and Design: The Hardware/Software Interface***", 5th Edition. (Module 3 & 4)
3. J. Hennessy and D. Patterson, "***Computer Architecture, A quantitative approach***", 5th Edition. (Module 3)
4. Hamacher, Vranesic & Zaky, "***Computer Organization***" (5th Ed), McGraw Hill. (Module 4 & 5)

References

1. William Stallings, "***Computer Organization and Architecture: Designing for Performance***", Pearson, 9/e, 2013.
2. R.P.Jain ,"***Modern Digital Electronics***", McGraw Hill.,Fourth Edition,2009
3. Mano, "***Digital Design : With an Introduction to Verilog HDL***", Pearson Education, 5th Edition (2014)

Course Contents and Lecture Schedule

No	**Topic**	**No. of Lectures**
	Module 1	**11**
1	Representation of signed numbers – 1's complement and 2's complement, Logic gates - AND, OR, NOT, NAND, NOR, XOR	2
1.1	Boolean algebra - Basic laws and theorems, Boolean functions - truth table.	2
1.2	Standard forms of Boolean Expressions – Sum of Products and Product of Sums - minimization of Boolean function using Karnaugh map method - Realization using logic gates.	2
1.3	Floating point numbers	1
1.4	Combinational Circuits - Half adder - Full Adder	2
1.5	Decoder – Encoder - Multiplexers – Demultiplexers	2
	Module 2	**10**
2.1	Sequential circuit - Clocking, Flip flops -RS – JK- D -T flip flops	3
2.2	Counters - Synchronous and asynchronous counters - UP/DOWN counters.	3
2.3	Registers - Serial in serial out - Serial in parallel out - Parallel in serial out - Parallel in parallel out registers	2
2.4	Introduction to arduino and raspberry pi	2
	Module 3	**10**
3.1	Computer abstractions and technology - Introduction, Computer architecture	4
3.2	Technologies for building processors and memory, Performance, instruction	4
3.3	Classifying instruction set architectures, Memory addressing, Encoding an	2
	Module 4	**9**
4.1	The Processor - Introduction, Logic design conventions, Building a datapath, A simple implementation scheme.	3
4.2	An Overview of pipelining - Pipelined datapath and control - Structural hazards - Data hazards - Control hazards	3
4.3	I/O organization - Accessing I/O devices, Interrupts - Handling multiple devices- Direct memory access	3

No	Topic	No. of Lectures
	Module 5	**8**
5.1	The memory system – basic concepts, semiconductor RAM memories, organization	2
5.2	Static and dynamic RAM, Structure of larger memories, semiconductor ROM memories, Speed, size and cost	2
5.3	Cache memory – mapping functions – replacement algorithms,	2
5.4	Virtual memory – paging and segmentation.	2

20MCA105	ADVANCED DATA STRUCTURES	CATEGORY	L	T	P	CREDIT
		GENERAL	3	1	0	4

Preamble: A graduate course in Computer Applications should give due exposure to the recent developments. Since Data structures is a central pillar of any program on Computer Science/ Applications, this course is designed to build upon the knowledge acquired at the undergraduate level and familiarise students with a bunch of modern data structures which are quite useful to solve, in the most effective manner, the modern, real life problems.

Prerequisite: Basic Data Structures

Course Outcomes: After the completion of the course the student will be able to

CO 1	Remember the Basic Data Structures and understand the Set Data Structure and its implementation.
CO 2	Understand Advanced Tree Structures for the design of efficient algorithms
CO 3	Understand Advanced Heap Structures suitable for solving Computational problems involving Optimisation and analysing these data structures using amortised analysis.
CO 4	Understand Advanced Graph algorithms suitable for solving advanced computational problems
CO 5	Understand the basic operation of Blockchaining along with the data structures used in it and the challenges in Blockchain data.

Mapping of course outcomes with program outcomes

	PO 1	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO 10**	**PO 11**	**PO 12**
CO 1	3	2	2		1							
CO 2	2	2	3	2	1		1					
CO 3	2	3	3	2	1		1					
CO 4	3	3	2	1	2		1					
CO 5	3	2	2	2	3		1					

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination
	1	2	
Remember	10	10	10
Understand	20	20	20
Apply	20	20	30
Analyse			
Evaluate			
Create			

Mark distribution

Total Marks	CIE	ESE	ESE Duration
100	40	60	3 hours

Continuous Internal Evaluation Pattern:

Attendance	: 8 marks
Continuous Assessment Test (2 numbers)	: 20 marks
Assignment/Quiz/Course project	: 12 marks

End Semester Examination Pattern: There will be two parts; Part A and Part B. Part A contains 10 compulsory short answer questions, 2 from each module. Each question carries 3 marks. Part B contains 2 questions from each module of which student should answer any one. Each question can have maximum 2 sub-divisions and carry 6 marks.

Course Level Assessment Questions

Course Outcome 1 (CO1):

(a) Review the basic data structures such as array, linked list, stack, queue etc.
(b) Understand the set data structure and its implementation
(c) Understand the Disjoint set data structure
(d) Learn the basics of Amortised Analysis and its important types

Course Outcome 2 (CO2)

(a) Understand Balanced Binary Search Trees and the idea of Rotations
(b) Understand Red Blak Trees and their operations
(c) Understand B Trees and operations
(d) Obtain a basic awareness of Splay Trees and Suffix Trees.

Course Outcome 3(CO3):

(a) Understand the concepts of Mergeable Heaps and their operations.
(b) Understand the Binomial Heaps and its operations along with their amortised analysis
(c) Understand the Fibonacci Heaps and its operations along with their amortised analysis

Course Outcome 4 (CO4):

(a) Understand Graphs traversal techniques and topological sorting using these
(b) Understand the algorithms for finding the strongly connected components and biconnected components in a graph.
(c) Understand the Prim's and Krusksl's algorithms and their implementation
(d) Understand the Dijikstra's Single Source Shortest path algorithm and implementing it using Advanced Heap Structures.

Course Outcome 5 (CO5):

(a) Understand a basic overview of the Blockchain system architecture.
(b) Understand the Blockchain Data Structures and Data Types.
(c) Understand the problems and challenges in Blockchain data.

Model Question paper

Reg No.:________________		Name:____ _
APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY FIRST SEMESTER M.C.A.DEGREE EXAMINATION, MODEL QUESTION PAPER		
Course Code: 20MCA105		
Course Name: ADVANCED DATA STRUCTURES		
Max. Marks: 60		Duration: 3 Hours
PART A		

		Answer all questions, each carries 3 marks.	Marks
1		What is meant by Hashing ?	(3)
2		How does Amortised Analysis differ from Average Case Analysis?	(3)
3		What is meant by Balanced Binary Search Tree? Give an example for a balanced binary search tree and an unbalanced one.	(3)
4		What is meant by Suffix Tree?	(3)
5		Give a valid Binomial heap with nodes 3,5,7,10,12,15.	(3)
6		Explain how fibonacci heaps are implemented?	(3)
7		What do you mean by Minimum Costs Spanning Tree?	(3)
8		What is meant by Strongly Connected Components? Illustrate with an example	(3)
9		What is meant by Block Chaining?	(3)
10		What is Contract Data ?	(3)
		PART B	
		Answer any one question from each module. Each question carries 6 marks.	
		Module I	
11		How do you perform Amortised Analysis using Accounting method? Illustrate with Multipop Stack example.	(6)
		OR	
12		Explain any three Hashing functions.	(6)
		Module II	
13		What is meant by Red Black Tree? Explain how insertion is done in a Red Black Tree.	(6)
		OR	
14		Give notes on B-Trees and Splay Trees.	(6)
		Module III	
15		Explain how the Decrease-Key operation is performed on Binomial Heaps. What is the Amortised Cost of this operation?	(6)
		OR	
16		Describe how the Delete-Key operation is performed in a Fibonacci heap? Illustrate with an example.	(6)
		Module IV	
17		Explain the Breadth First Search algorithm with a suitable example.	(6)
		OR	
18		Explain the Prim’s algorithm with an example.	(6)

Module V			
19		Explain the Blockchain architecture in detail.	(6)
OR			
20		Explain the problems to be solved in Blockchain Data Analysis.	(6)
****			

Syllabus

Module 1 [12 hrs]

Review of basic data structures- Array, linked list and its variants, Stack ,Queue and Trees
Set Data Structure:- Representation of sets, Set implementation using bit string.
Hashing :- SImple hash functions, Collision and Collision Resolution techniques
Amortised Analysis - Aggregate, Accounting and Potential Methods (using the examples Multipop Stack and Incrementing Binary Counter only)
Disjoint sets- representations, Union, Find algorithms

Module 2 [10 hrs]

Advanced Tree Structures:- Balanced Binary Search trees, Red-Black trees- Properties of Red Black trees, Rotations, Insertion, Deletion. B-Trees- Basic operations on B-Trees – Insertion and Deletion, Introduction to Splay Trees and Suffix Trees

Module 3 [10 hrs]

Advanced Heap Structures:- Mergeable Heaps and operations on Mergeable Heaps. Binomial Heaps, Binomial Heap operations and Analysis, Fibonacci Heaps, Fibonacci Heap operations and Analysis**.**

Module 4 [14 hrs]

Advanced Graph Structures : Representation of graphs, Depth First and Breadth First Traversals, Topological Sorting, Strongly connected Components and Biconnected Components Minimum Cost Spanning Tree algorithms- Prim’s Algorithm, Kruskal’ Algorithm,. Shortest Path Finding algorithms – Dijikstra’s single source shortest paths algorithm

Module 5[8 hrs]

Blockchain Data Structure:- Blockchain Architecture, Blockchain Data Structures and Data types, Contract Data, Problems to be solved in Blockchain data analysis

Text Books

1. Cormen T.H., Leiserson C.E, Rivest R.L. and Stein C, ***Introduction to Algorithms***, Prentice Hall India, New Delhi, 2004 [Modules 1 to 4]

2. Yang, Xiaojing, Jinshan Liu, and Xiaohe Li. "***Research and Analysis of Blockchain Data." Journal of Physics: Conference Series. Vol. 1237***. No. 2. IOP Publishing, 2019.

Reference Books

1. Kleinberg, Jon, and Eva Tardos. ***Algorithm design***. Pearson Education India, 2006.

2. Aho A.V., Hopcroft J.E., and Ullman J.D., ***Data Structures and Algorithms***, Pearson Education, New Delhi, 1983.

3. Sahni S., ***Data Structures, Algorithms, and Applications in C++,*** Mc Graw Hill, Singapore, 1998.

Course Contents and Lecture Schedule

No	Topic	No. of Lecture Hours
1	**Review of basic data structures**	**10Hrs**
1.1	Array, Stack and Queue	
1.2	Linked list and its variants	
1.3	Representation of sets, Set implementation using bit string.	
1.4	Hashing – SImple hash functions	
1.5	Collision and Collision Resolution techniques	
1.6	Amortised Analysis	
1.7	Aggregate Method (Multipop Stack and Incrementing Binary Counter)	
1.8	Accounting Method (Multipop Stack and Incrementing Binary Counter)	
1.9	Potential Method (Multipop Stack and Incrementing Binary Counter)	
1.10	Disjoint sets- representations	
1.11	Union, Find algorithms	
2	**Advanced Tree Structures**	**10Hrs**
2.1	Balanced Binary Search trees	
2.2	Red-Black trees	
2.3	Properties of Red Black trees	
2.4	Rotations	
2.5	Insertion	

2.6	Deletion	
2.7	B-Trees	
2.8	Insertion and Deletion	
2.9	Splay Trees	
2.10	Suffix Trees	
3	**Advanced Heap Structures**	**8Hrs**
3.1	Mergeable Heaps	
3.2	Operations on Mergeable Heaps	
3.3	Binomial Heaps	
3.4	Binomial Heaps operations and Analysis	
3.5	Fibonacci Heaps	
3.6	Fibonacci Heap operations and Analysis.	
4	**Advanced Graph Structures**	**12Hrs**
4.1	Representation of graphs	
4.2	Depth First and Breadth First Traversals	
4.3	Topological Sorting	
4.4	Strongly connected Components	
4.5	Biconnected Components	
4.6	Minimum Cost Spanning Tree	
4.7	Prim's Algorithm	
4.8	Kruskal’s Algorithm	
4.9	Dijikstra’s single source shortest paths algorithm	
5	**Blockchain Data Structure**	**8Hrs**
5.1	Blockchain Architecture	
5.2	Blockchain Data Structures	
5.3	Blockchain Data types	
5.4	Contract Data	
5.5	Problems to be solved in Blockchain data analysis	

20MCA107	ADVANCED SOFTWARE ENGINEERING	CATEGORY	L	T	P	CREDIT
		GENERAL	3	1	0	4

Preamble:
Most of the programs on Computer Applications do not give due importance to teach Software Engineering in an Industry perspective. But this course, built upon the tools and techniques prevalent in Industry today, is supposed to make students Industry-ready.

Prerequisite: Programming proficiency in at least one of C, C++, Java, Python or PHP programming languages.

Course Outcomes: After the completion of the course the student will be able to

CO 1	Get a full view of the Software life cycle
CO 2	Gain a deep knowledge of Software Planning, Analysis and Design and Software Engineering Models
CO 3	Have a great comprehension of Coding Practices, Version Control using ‘git’ and Software Quality
CO 4	Acquire ample grasp of Design Patterns
CO 5	Get deeply familiarised with Software Testing and its automation
CO 6	Start using Agile Methodology
CO 7	Begin to apply CI/CD techniques in Software development

Mapping of course outcomes with program outcomes

	PO 1	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO 10**	**PO 11**	**PO 12**
CO 1		2	2					3			1	1
CO 2		3	3					3				
CO 3					3				3	2	2	
CO 4			3		3							
CO 5					3					2	3	
CO 6					2			2	2		2	3
CO 7					3			1		2		

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination
	1	2	
Remember	10	10	10
Understand	20	20	20
Apply	10	10	20
Analyse			
Evaluate			
Create	10	10	10

Mark distribution

Total Marks	CIE	ESE	ESE Duration
100	40	60	3 hours

Continuous Internal Evaluation Pattern:

Attendance	: 8 marks
Continuous Assessment Test (2 numbers)	: 20 marks
Assignment/Quiz/Course project	: 12 marks

End Semester Examination Pattern: There will be two parts; Part A and Part B. Part A contains 10 compulsory short answer questions, 2 from each module. Each question carries 3 marks. Part B contains 2 questions from each module of which student should answer any one. Each question can have maximum 2 sub-divisions and carry 6 marks.

Course Level Assessment Questions

Course Outcome 1 (CO1):

(a) Understand the software development as an engineering process and its stages.

(b) Understand Software development lifecycle (SDLC).

(c) Understand software engineering models.

(d) Learn how to prepare software requirements specification, approaches and methodologies to prepare requirement specifications document.

Course Outcome 2 (CO2)

(a) Understand writing industry-grade software programs, following style guides and coding standards.

(b) Learn core concepts of software version control system and common operations with Git distributed version control system.

(c) Understanding software quality concepts with respect to software requirement specifications document, what to conform to at various stages of SDLC.

(d) Understand what to ensure at various stage of SDLC to ensure quality of developed software system.

Course Outcome 3(CO3):

(a) Learn Object Oriented Programming concepts comprehensively.

(b) Learn the concept of Design Patterns, category of patterns, and how to select appropriate design patterns.

(c) Understand Unit testing concepts and xUnit architecture.

(d) Learn Unit testing frameworks and writing unit testing for Java and one of PHP or Python.

(e) Understand the concepts Continuous Integration and Continuous Delivery (CICD).

Course Outcome 4 (CO4):

(a) Knowledge of Git distributed version control system to use in a product environment.

(b) Knowledge of OOP paradigm and software Design Patterns to design the software system.

(c) Knowledge of unit testing frameworks such as Junit, uniitest, phpdbg for wiring units tests in a software production environment.

(d) Knowledge of software testing CI/CD practices.

Course Outcome 5 (CO5):

(a) Understand software testing concepts and principles.

(b) Learn common approaches to ensure software quality through testing.

(c) In-depth understanding of various types of testing methodologies.

(d) Learn about testing automation and understand commonly used test automation types.

(e) Learn to use Robot framework.

Course Outcome 6 (CO5):

(a) Understand the concepts of Agile methodology.

(b) Learn to use Scrum framework for implementing Agile methodology for executing a software development process.

(c) Learn to monitor a software development project using a Scrum tool.

Course Outcome 7 (CO5):

(a) Understand the concepts of Software Configuration Management.

(b) Learn about build and deployment environments.

(c) Understand the concepts of Continuous Integration and essential practices.

(d) Understand the concepts of deployment automation and learn to use Ansible.

Model Question paper

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY
First Semester MCA Degree Examination (R&S)

Course Code: 20MCA107
Course Name: ADVANCED SOFTWARE ENGINEERING

Total Marks: 60 Duration: 3 Hours

PART A
Answer all questions, each carries3 marks. Marks

1. Why is Software Engineering important? (3)
2. What are the desired requirements of a good software engineering model? (3)
3. What is the purpose of a version control system? (3)
4. Explain the different ways to fix commits in Git (3)
5. What is anti-pattern? (3)
6. What is an abstract test? (3)
7. Distinguish between black box testing and white box testing. (3)
8. Draw a model Sprint Backlog for the login module of a simple web portal (3)

9. Write a short note on release candidate (3)

10. Differentiate continuous delivery and continuous deployment (3)

PART B

Required to answer one question from each module in full.

Each module carries 6 marks for either of the questions.

Module I

11. Prepare a basic Software Requirements Specification for Savings Bank accounts. (6)

OR

12. How is Use Case different from User Stories? Enlist the advantage of each. (6)

Module II

13. How do you create, switch and view branches in Git? explain how to merge commits between branches. (6)

OR

14. You have cloned a repository which was then modified by another developer. You make changes locally and try to execute push. What are the possible outputs? How will you solve the problems, if any? (6)

Module III

15. Explain the important design patterns. (6)

OR

16. When are assertions and expected error tests used in Unit tests? (6)

Module IV

17. Write down the scrum. (6)

OR

18. Differentiate Black box testing and White box testing. Give appropriate example for each for "only black box testing is possible" and "necessary to do white box testing" scenarios. (6)

Module V

19. Explain the strategy for implementing Continuous integration. (6)

OR

20. What is a deployment pipeline? Explain the anatomy of a deployment pipeline with a (6) neat diagram. Comment on the various stages of a deployment pipeline.

Syllabus

Module 1 [8 hrs]

Introduction to Software Engineering: What is Software Engineering, Characteristics of Software.

Life cycle of a software system: software design, development, testing, deployment, Maintenance.

Project planning phase: project objectives, scope of the software system, empirical estimation models, COCOMO, staffing and personnel planning.

Software Engineering models: Predictive software engineering models, model approaches, prerequisites, predictive and adaptive waterfall, waterfall with feedback (Sashimi), incremental waterfall, V model; Prototyping and prototyping models.

Software requirements specification, Eliciting Software requirements, Requirement specifications, Software requirements engineering concepts, Requirements modelling, Requirements documentation. Use cases and User stories.

Module 2 [10 hrs]

Programming Style Guides and Coding Standards; Literate programming and Software documentation; Documentation generators, Javadoc, phpDocumentor.

Version control systems basic concepts; Concept of Distributed version control system and Git; Setting up Git; Core operations in Git version control system using command line interface (CLI): Clone a repository; View history; Modifying files; Branching; Push changes, Clone operation, add, commit, log, diff commands, conflict resolution. Pushing changes to the master; Using Git in IDEs and UI based tools.

Software Quality: Understanding and ensuring requirements specification quality, design quality, quality in software development, conformance quality.

Module 3 [10 hrs]

OOP Concepts; Design Patterns: Basic concepts of Design patterns, How to select a design pattern, Creational patterns, Structural patterns, Behavioural patterns. Concept of Anti-patterns.

Unit testing and Unit Testing frameworks, The xUnit Architecture, Writing Unit Tests using at least one of Junit (for Java), unittest (for Python) or phpdbg (PHP). Writing tests with Assertions, defining and using Custom Assertions, single condition tests, testing for expected errors, Abstract test.

Module 4 [10 hrs]

Concepts of Agile Development methodology; Scrum Framework.

Software testing principles, Program inspections, Program walkthroughs, Program reviews; Blackbox testing: Equivalence class testing, Boundary value testing, Decision table testing, Pairwise testing, State transition testing, Use-case testing; White box testing: control flow testing, Data flow testing.

Testing automation: Defect life cycle; Regression testing, Testing automation; Testing non-functional requirements.

Module 5[10 hrs]

Software Configuration Management: Using version control, Managing dependencies, Managing software configuration, Managing build and deployment environments.

Continuous Integration: Prerequisites for continuous integration, Essential practices.

Continuous Delivery: Principles of Software delivery, Introduction and concepts.

Build and deployment automation, Learn to use Ansible for configuration management.

Test automation (as part of continuous integration), Learn to set up test automation cases using Robot Framework.

Notes

1. At the end of Module 1, conduct the following class work with appropriate evaluation points: Prepare Software Specification Document for a moderately complex process flow system (*e.g.* Broadband fault booking and resolution system covering technical, operational and commercial aspects, covering organizational and subscriber use cases).

2. At the end of Module 2, clone an open source project using Git and perform all based operations.

Reference Books

1. Philip A. Laplante, *What Every Engineer Should Know about Software Engineering*, CRC Press [Module 1]
2. Murali Chemuturi, *Mastering Software Quality Assurance: Best Practices, Tools and Technique for Software Developers*, J Ross Publishing [Module 2]
3. Ben Straub, Scott Chacon, *Pro Git*, 2nd Edition, Apress [Module 2]
4. Erich Gamma et. al., *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley [Moule 3]
5. Vaskaran Sarcar, *Java Design Patterns: A Hands-On Experience with Real-World Examples*, Apress [Module 3]
6. Alistair Cockburn and Robert Cecil Martin, *Agile Software Development: The Cooperative Game (2nd edition)*, Addition Wesley [Module 4]

7. Ken Schwaber , *Agile Software Development with Scrum*, Pearson [Module 4]

8. Lisa Crispin, Agile Testing: *A Practical Guide for Testers and Agile Teams*, Adison Wesley

9. Paul Hamill, *Unit Test Frameworks*, O'Reilly Media [Module 4]

10. Glenford J. Myers, et. al., *The Art of Software Testing*, Wiley [Module 4, 5]

11. Lee Copeland, *A Practitioner's Guide to Software Test Design*, Artech House Publishers [Module 4, 5]

12. Jez Humble and David Farley, *Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation*, Pearson Education [Module 5]

Web-based Resources

1. *Git Handbook* https://guides.github.com/introduction/git-handbook/ Retrieved 8 July 2020 [Module 2]

2. *Git User Manual* https://mirrors.edge.kernel.org/pub/software/scm/git/docs/user-manual.html Retrieved 8 July 2020 [Module 2]

3. *Introduction to Software Engineering/Quality* https://en.wikibooks.org/wiki/Introduction_to_Software_Engineering/Quality Retrieved 8 July 2020 [Module 2]

4. *Understanding software design patterns* https://opensource.com/article/19/7/understanding-software-design-patterns Retrieved 8 July 2020 [Module 3]

5. *The Scrum Guide* https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf Retrieved 8 July 2020 [Module 4]

6. *unittest — Unit testing framework* https://docs.python.org/3/library/unittest.html Retrieved 8 July 2020 [Module 4]

7. What is CI/CD? https://www.redhat.com/en/topics/devops/what-is-ci-cd Retrieved 8 July 2020 [Module 5]

Course Contents and Lecture Schedule

No	Topic	No. of Lecture Hours
1	**Software Engineering**	
1.1	What is Software Engineering, Characteristics of Software Engineering	1
1.2	Life cycle of a software system	1
1.3	Project planning	1
1.4	Software Engineering Models	2
1.5	Software Requirements Specification	3
2	**Industry Best Practices**	
2.1	Programming style guides and coding standards	1
2.2	Software version control systems, basic concepts	1
2.3	Git distributed version control system, introduction	2
2.4	Common operations in Git	4

No	Topic	No. of Lecture Hours
2.5	Software quality, achieving	2
3	**System Design Methodologies**	
3.1	Object Oriented Programming	1
3.2	Software Design Patterns	4
3.3	Unit Testing concepts and xUnit architecture	1
3.4	Unit testing frameworks: Junit, unittest, phpdbg	2
3.5	Writing unit test code	2
4	**Agile Development Methodology**	
4.1	Agile Development methodology, introduction	2
4.2	Scrum framework	5
4.3	Automated testing	3
5	**Continuous Integration and Continuous Development (CI/CD)**	
5.1	Configuration Management	2
5.2	Continuous Integration, concepts and practices	2
5.3	Continuous Delivery, concepts and practices	2
5.4	Build and deployment automation	2
5.5	Test automation for CI/CD	2

20MCA131	PROGRAMMING LAB	CATEGORY	L	T	P	CREDIT
		PRACTICAL	0	1	3	2

Preamble: This course introduces a basic step towards program writing and develops the logical ability and problem-solving skill using Python Programming Language. Students are able to do testing and debugging of code written in Python.

Prerequisite: None

Course Outcomes: After the completion of the course the student will be able to

CO 1	Understands basics of Python Programming language including input/output functions, operators, basic and collection data types
CO 2	Implement decision making, looping constructs and functions
CO 3	Design modules and packages - built in and user defined packages
CO 4	Implement object-oriented programming and exception handling.
CO 5	Create files and form regular expressions for effective search operations on strings and files.

Mapping of course outcomes with program outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	2	2	2	1	2							
CO 2	3	3	3	2	2							
CO 3	3	3	3	3	3						1	
CO 4	3	3	3	3	3						1	
CO 5	3	3	3	3	3						1	

Assessment Pattern

Bloom's Category	**Continuous Assessment Tests**		**End Semester Examination**
	1	**2**	
Remember (K1)			
Understand (K2)			
Apply (K3)	10	10	10
Analyse (K4)	10	10	10
Evaluate (K5)	10	10	10
Create(K6)	20	20	20

Mark distribution

Total Marks	CIE	ESE	ESE Duration
100	50	50	3 hours

Continuous Internal Evaluation Pattern:

Maximum Marks: 50	
Attendance	15%
Maintenance of daily lab record and GitHub management	20%
Regular class viva	15%
Timely completion of day to day tasks	20%
Tests/Evaluation	30%

End Semester Examination Pattern:

Maximum Marks: 50			
Verification of Daily program record and Git Repository			5 marks
Viva			10 marks
Problem solving (Based on difficulty level, one or more questions may be given)	Flowchart / Algorithm / Structured description of problem to explain how the problem can be solved / Interface Design	15%	35 marks
	Program correctness	50%	
	Code efficiency	15%	
	Formatted output and Pushing to remote Git repository	20%	
Total Marks			50 marks

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Familiarizing Text Editor, IDE, Code Analysis Tools etc // Use any IDE like PyCharm, PyDev…
2. Display future leap years from current year to a final year entered by user.
3. List comprehensions:
   (a) Generate positive list of numbers from a given list of integers
   (b) Square of N numbers
   (c) Form a list of vowels selected from a given word
   (d) List ordinal value of each element of a word (Hint: use ord() to get ordinal values)
4. Count the occurrences of each word in a line of text.
5. Prompt the user for a list of integers. For all values greater than 100, store 'over' instead.
6. Store a list of first names. Count the occurrences of 'a' within the list
7. Enter 2 lists of integers. Check (a) Whether list are of same length (b) whether list sums to same value (c) whether any value occur in both
8. Get a string from an input string where all occurrences of first character replaced with '$', except first character.
   [eg: onion -> oni$n]
9. Create a string from given string where first and last characters exchanged. [eg: python -> nythop]
10. Accept the radius from user and find area of circle.
11. Find biggest of 3 numbers entered.
12. Accept a file name from user and print extension of that.
13. Create a list of colors from comma-separated color names entered by user. Display first and last colors.
14. Accept an integer n and compute n+nn+nnn.
15. Print out all colors from color-list1 not contained in color-list2.
16. Create a single string separated with space from two strings by swapping the character at position 1.
17. Sort dictionary in ascending and descending order.

18. Merge two dictionaries.
19. Find gcd of 2 numbers.
20. From a list of integers, create a list removing even numbers.

Course Outcome 2 (CO2)

1. Program to find the factorial of a number
2. Generate Fibonacci series of N terms
3. Find the sum of all items in a list
4. Generate a list of four digit numbers in a given range with all their digits even and the number is a perfect square.
5. Display the given pyramid with step number accepted from user.

```
Eg: N=4
1
2  4
3  6  9
4  8 12  16
```

6. Count the number of characters (character frequency) in a string.
7. Add ‘ing’ at the end of a given string. If it already ends with ‘ing’, then add ‘ly’
8. Accept a list of words and return length of longest word.
9. Construct following pattern using nested loop

```
*
* *
* * *
* * * *
* * * * *
* * * *
* * *
* *
*
```

10. Generate all factors of a number.

11. Write lambda functions to find area of square, rectangle and triangle.

Course Outcome 3(CO3):

1. Work with built-in packages
2. Create a package graphics with modules rectangle, circle and sub-package 3D-graphics with modules cuboid and sphere. Include methods to find area and perimeter of respective figures in each module. Write programs that finds area and perimeter of figures by different importing statements. (Include selective import of modules and import * statements)

Course Outcome 4 (CO4):

1. Create Rectangle class with attributes length and breadth and methods to find area and perimeter. Compare two Rectangle objects by their area.
2. Create a Bank account with members account number, name, type of account and balance. Write constructor and methods to deposit at the bank and withdraw an amount from the bank.
3. Create a class Rectangle with private attributes length and width. Overload '<' operator to compare the area of 2 rectangles.
4. Create a class Time with private attributes hour, minute and second. Overload '+' operator to find sum of 2 time.

5. Create a class Publisher (name). Derive class Book from Publisher with attributes title and author. Derive class Python from Book with attributes price and no_of_pages. Write a program that displays information about a Python book. Use base class constructor invocation and method overriding.

Course Outcome 5 (CO5):

1. Write a Python program to read a file line by line and store it into a list.
2. Python program to copy odd lines of one file to other
3. Write a Python program to read each row from a given csv file and print a list of strings.
4. Write a Python program to read specific columns of a given CSV file and print the content of the columns.
5. Write a Python program to write a Python dictionary to a csv file. After writing the CSV file read the CSV file and display the content.

Syllabus:

Input, Output and Import Functions, Operators, Data Types, Decision Making & Loops, Functions, Modules and Packages, File Handling, Object Handling, Exception Handling, Regular Expressions

Reference Books

1. Wesley J. Chun, "***Core Python Applications Programming***", 3rd Edition , Pearson Education, 2016

2. Charles Dierbach, "***Introduction to Computer Science using Python***", Wiley, 2015

3. Jeeva Jose, "***Taming Python by Programming***", Khanna Publishers, New Delhi, 2018

4. Downey, A. et al., "***How to think like a Computer Scientist: Learning with Python***", John Wiley, 2015

Web References

1. https://archive.org/details/MIT6.00SCS11
2. https://www.coursera.org/course/pythonlearn
3. http://www.learnerstv.com/Free-Computer-Science-Video-lectures-ltv163-Page1.htm
4. https://www.coursera.org/learn/python-databases

Course Contents and Lab Schedule

Topic	No. of hours
1. Input, Output and Import Functions	3
2. Operators	5
3. Data Types	6
4. Decision Making & Loops	6
5. Functions	5
6. Modules and Packages	6
7. File Handling	5
8. Object Handling	5
9. Exception Handling	2
10. Regular Expressions	4

20MCA133	WEB PROGRAMMING LAB	CATEGORY	L	T	P	CREDIT
		PRACTICAL	0	1	3	2

Preamble: With a dynamic learn-by-doing focus, this laboratory course encourages the students to explore the designing of web application by implementing the relevant and recent techniques. This course challenges the students to exercise their creativity in both programming and designing.

Prerequisite: Basic understanding of computer programming, Internet and Database etc. is very helpful.

Course Outcomes: After the completion of the course the student will be able to;

CO 1	Explore markup languages features and create interactive web pages using them.
CO 2	Learn and design client-side validation using scripting languages.
CO 3	Design front end web page and connect to the back-end databases.
CO 4	Do Client-side & Server-side scripting
CO 5	Develop Web Applications

Mapping of course outcomes with program outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	3	3	3	2	2		3	3				
CO 2	3	3	3	2	2		3	3	1			
CO 3	3	3	3	2	2		3	3				
CO 4	3	3	3	2	2		3	3				2
CO 5	3	3	3	3	3		3	3			2	2

Assessment Pattern

Bloom's Category	**Continuous Assessment Tests**		**End Semester Examination**
	1	**2**	
Remember(K1)			
Understand(K2)			
Apply(K3)	10	10	10
Analyse(K4)	10	10	10

Evaluate(K5)	10	10	10
Create(K6)	20	20	20

Mark distribution

Total Marks	CIE	ESE	ESE Duration
100	50	50	3 hours

Continuous Internal Evaluation Pattern:

Maximum Marks: 50	
Attendance	15%
Maintenance of daily lab record and GitHub management	20%
Regular class viva	15%
Timely completion of day to day tasks	20%
Tests/Evaluation	30%

End Semester Examination Pattern:

Maximum Marks: 50			
Verification of Daily program record and Git Repository			5 marks
Viva			10 marks
Problem solving (Based on difficulty level, one or more questions may be given)	Flowchart / Algorithm / Structured description of problem to explain how the problem can be solved / Interface Design	15%	35 marks
	Program correctness	50%	
	Code efficiency	15%	
	Formatted output and Pushing to remote Git repository	20%	
Total Marks			50 marks

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Model a simple HTML file to demonstrate the use of different tags. (K3)
2. Create a HTML file to link to different HTML page which contains images, tables, and also link within a page. (K6)
3. Create a HTML page with different types of frames such as floating frame, navigation frame & mixed frame. (K6)
4. Analyze CSS by applying the different styles using inline, external & internal style sheets in a HTML file. (K4)
5. Demonstrate a registration form using HTML. (K3)

Course Outcome 2 (CO2)

1. Create a HTML page to explain the use of various predefined functions in a string and math object in java script. (K6)
2. Generate the calendar using JavaScript code by getting the year from the user. (K6)
3. Create a HTML registration form and to validate the form using JavaScript code. (K6)
4. Evaluating JavaScript Event Handling for every click of a button to change the background color of a HTML page. (K5)
5. Create a HTML page to display a new image and text when the mouse comes over the existing content in the page using JavaScript Event Handling. (K6)
6. Create a HTML page to show online exam using JavaScript. (K6)

Course Outcome 3(CO3):

1. Develop a PHP program to connect to a database and retrieve data from a table and show the details in a neat format. (K6)

Course Outcome 4 (CO4):

1. Outline a registration form using PHP and do necessary validations. (K4)
2. Compose Electricity bill from user input based on a given tariff using PHP. (K6)

3. Build a PHP code to store name of students in an array and display it using print_r function. Sort and Display the same using asort & arsort functions. (K6)
4. Build a PHP code to store name of Indian Cricket players in an array and display the same in HTML table. (K6)

Course Outcome 5 (CO5):

1. Develop Web applications using HTML and PHP and deploy. (K6)
2. Using PHP and MySQL, develop a program to accept book information viz. Accession number, title, authors, edition and publisher from a web page and store the information in a database and to search for a book with the title specified by the user and to display the search results with proper headings. (K6)
3. Develop a web application for Airline Reservation System using any PHP framework (Laravel, CodeIgniter, Symfony, CakePHP etc.). (K6)
4. Test the application on an Application Server. (K5)

Syllabus

Introduction To Web: Client/Server concepts, Components of Web Application, Types of Web Content, Overview of HTTP - HTTP request – response, Generation of dynamic web pages, Application Servers, Web Security.

Markup Language (HTML): Formatting and Fonts, Commenting Code, Anchors, Backgrounds, Images, Hyperlinks, Lists, Tables, Frames, HTML Forms.

Cascading Style Sheet (CSS): The need for CSS, Basic syntax and structure, Inline Styles, Embedding Style Sheets, Linking External Style Sheets, Backgrounds, Manipulating text, Margins and Padding, Positioning using CSS.

Client Side Scripting using JavaScript: Core features, Data types and Variables, Operators -Expressions and Statements, Functions, Objects, Array, String - Date and Math related Objects, Document Object Model, Event Handling, Form handling and validations.

An overview of Relational Database Design: Tables, Attributes, Tuples, Primary keys, Foreign keys, Indexes, DDL Commands – CREATE, ALTER, DROP and TRUNCATE; DML Commands – SELECT, INSERT, UPDATE and DELETE.

Server Side Scripting using PHP: Setting up the environment (Example - XAMP server), PHP Programming basics - Print/echo, Variables and constants, Strings and Arrays, Operators, Control structures and looping structures, Functions, Reading Data in Web Pages, Embedding PHP within HTML, Establishing connectivity with database, Debugging with phpdbg.

Web Application development in any PHP framework (Laravel, CodeIgniter, Symfony, CakePHP etc.): Naming convention, MVC model, Connectivity with Database, Database interaction.

Debugging web apps: Browser debugging tools (Any browser web developer tools) - View and change the DOM and CSS, Console, Debug JavaScript, View and debug network activity, Performance tools etc.

Reference Books

1. David Flanagan, "***JavaScript: The Definitive Guide***", 6th Edition", O'Reilly Media
2. Douglas E Comer, "***The Internet Book: Everything You Need to Know About Computer Networking and How the Internet Works***", 4th Edition, Prentice Hall
3. Harvey Deitel and Abbey Deitel, "***Internet and World Wide Web - How To Program***", 5th Edition,

   Pearson Education
4. Abraham Silberschatz, Henry F. Korth, S. Sudarshan, "***Database System Concepts***", McGraw Hill Education, 6th Edition (2011)
5. Steve Suehring, Tim Converse, and Joyce Park, "***PHP6 and MySQL Bible***", Wiley India Pvt Ltd (2009)
6. Steven Holzner, "***PHP-The Complete Reference***", Tata McGraw Hill, 1st Edition (2007)
7. Thomas A Powell, Fritz Schneider, "***JavaScript: The Complete Reference***", 3rd Edition, Tata McGraw Hill

Web Resources

1. http://php.net/manual/
2. https://pepa.holla.cz/wp-content/uploads/2016/08/JavaScript-The-Definitive-Guide-6th-Edition.pdf
3. http://index-of.es/PHP/PHP6%20and%20MySQL%20Bible.pdf
4. https://www.udemy.com/course/html5-fundamentals-for-beginners/
5. https://www.udemy.com/course/programming-in-javascript/
6. https://www.udemy.com/course/php-mysql-tutorial/

List of Lab Experiments/Exercises

1. Create a simple HTML file to demonstrate the use of different tags.
2. Create a HTML file to link to different HTML page which contains images, tables, and also link within a page.
3. Create a HTML page with different types of frames such as floating frame, navigation frame & mixed frame.
4. Create a HTML file by applying the different styles using inline, external & internal style sheets.
5. Create a registration form using HTML.
6. Create a HTML page to explain the use of various predefined functions in a string and math object in java script.
7. Generate the calendar using JavaScript code by getting the year from the user.
8. Create a HTML registration form and to validate the form using JavaScript code.
9. Create a HTML page to change the background color for every click of a button using JavaScript Event Handling.
10. Create a HTML page to display a new image and text when the mouse comes over the existing content in the page using JavaScript Event Handling.
11. Create a HTML page to show online exam using JavaScript.
12. Develop a registration form using PHP and do necessary validations.
13. Compose Electricity bill from user input based on a given tariff using PHP.
14. Build a PHP code to store name of students in an array and display it using print_r function. Sort and Display the same using asort & arsort functions.
15. Build a PHP code to store name of Indian Cricket players in an array and display the same in HTML table.
16. Develop a PHP program to connect to a database and retrieve data from a table and show the details in a neat format.
17. Develop Web applications using HTML and PHP and deploy.
18. Using PHP and MySQL, develop a program to accept book information viz. Accession number, title, authors, edition and publisher from a web page and store the information in a database and to search for a book with the title specified by the user and to display the search results with proper headings.

19. Develop a web application for Airline Reservation System using any PHP framework (Laravel, CodeIgniter, Symfony, CakePHP etc.).
20. Test the application on an Application Server.

Note: *Students can be given a group micro project, so that they learn to work in a team environment. They can also be trained on project management tools.*

Course Contents and Lecture Schedule

Topic	No. of lectures
Client/Server concepts, Components of Web Application, Types of Web Content, Overview of HTTP - HTTP request – response, Generation of dynamic web pages, Application Servers, Web Security.	1Hr.
HTML - Formatting and Fonts, Commenting Code, Anchors,Backgrounds, Images, Hyperlinks.	4 Hrs.
HTML - Lists, Tables, Frames, HTML Forms.	4 Hrs.
The need for CSS, Basic syntax and structure, Inline Styles, Embedding Style Sheets, Linking External Style Sheets, Backgrounds.	4 Hrs.
CSS - Manipulating text, Margins and Padding, Positioning using CSS.	4 Hrs.
JavaScript: Core features, Data types and Variables, Operators - Expressions and Statements.	3 Hrs.
JavaScript: Functions, Objects, Array, String - Date and Math related Objects, Document Object Model, Event Handling.	4 Hrs.
JavaScript: Form handling and validations.	4 Hrs.
An overview of Relational Database Design: Tables, Attributes, Tuples, Primary keys, Foreign keys, Indexes, DDL Commands – CREATE, ALTER, DROP and TRUNCATE.	4 Hrs.
DML Commands – SELECT, INSERT, UPDATE and DELETE.	4 Hrs.
PHP: Setting up the environment (Example - XAMP server), PHP Programming basics - Print/echo, Variables and constants.	4 Hrs.
Strings and Arrays, Operators, Control structures and looping structures.	4 Hrs.
Functions, Reading Data in Web Pages, Embedding PHP within HTML, Establishing connectivity with database.	4 Hrs.
PHP framework: naming convention, MVC model, Connectivity with Database, Database Interaction.	6 Hrs.

20MCA135	**DATA STRUCTURES LAB**	**CATEGORY**	**L**	**T**	**P**	**CREDIT**
		PRACTICAL	**0**	**1**	**3**	**2**

Preamble: This is the companion course of 20MCA105 Advanced Data Structures and provides the students hands-on experience of the advanced data structures which will boost up the knowledge and confidence of students in applying these techniques while dealing with real life computing problems.

Prerequisite: Basic Data Structures, Knowledge of any programming language, preferably ‘C’.

Course Outcomes: After the completion of the course the student will be able to

CO 1	Use Debuggers, Profilers and advanced Compiler options.
CO 2	Implement the Set and Disjoint Set Data Structures.
CO 3	Understand the practical aspects of Advanced Tree Structures.
CO 4	Realise Modern Heap Structures for effectively solving advanced Computational problems.
CO 5	Implement Advanced Graph algorithms suitable for solving advanced computational problems.

Mapping of course outcomes with program outcomes

	PO 1	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO 10**	**PO 11**	**PO 12**
CO 1	2				3		1					
CO 2	**3**	2	2		1							
CO 3	**2**	2	3	2	1		1					
CO 4	**2**	3	3	2	1		1					
CO 5	**3**	3	2	1	2		1					

Assessment Pattern

Bloom’s Category	Continuous Assessment Tests		End Semester Examination
	1	2	
Remember(K1)			
Understand(K2)			
Apply(K3)	10	10	10
Analyse(K4)	10	10	10
Evaluate(K5)	10	10	10
Create(K6)	20	20	20

Mark distribution

Total Marks	CIE	ESE	ESE Duration
100	50	50	3 hours

Continuous Internal Evaluation Pattern:

Maximum Marks: 50	
Attendance	15%
Maintenance of daily lab record and GitHub management	20%
Regular class viva	15%
Timely completion of day to day tasks	20%
Tests/Evaluation	30%

End Semester Examination Pattern:

Maximum Marks: 50			
Verification of Daily program record and Git Repository			5 marks
Viva			10 marks
Problem solving (Based on difficulty level, one or more questions may be given)	Flowchart / Algorithm / Structured description of problem to explain how the problem can be solved / Interface Design	15%	35 marks
	Program correctness	50%	
	Code efficiency	15%	
	Formatted output and Pushing to remote Git repository	20%	
Total Marks			50 marks

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Write a C program ‘sum.c’ to add two numbers. Read the input from Standard Input and write output to Standard output. Compile and generate sum.out which is then debug with gdb.
2. Modify ‘sum.c’ by adding a function for finding the sum of two numbers. Then profile the executable with gprof.

Course Outcome 2 (CO2)

1. Create the Set ADT with Add, Remove, Union, Intersection and Difference operations. Implement using Bit Strings.
2. Implement the Disjoint set ADT with Create, Union and Find operations.
3. Implement Kruskal’s algorithm using Disjoint sets.

Course Outcome 3(CO3):

1. Implement B-Tree and its operations..
2. Implement Red Black Tree and the associated operations.

Course Outcome 4 (CO4):

1. Create the Binomial Heap ADT and implement the basic operations.
2. Use any Mergeable Heap to implement Single source shortest path algorithm.

Course Outcome 5 (CO5):

1. Finding the strongly connected components of a directed graph.
2. Prim's Algorithm for Minimum cost spanning tree.

Syllabus:

Based on the syllabus of 20MCA105 Advanced Data Structures.

Text Books

1. Cormen T.H., Leiserson C.E, Rivest R.L. and Stein C, ***Introduction to Algorithms***, Prentice Hall India, New Delhi, 2004

Reference Books

1. Kleinberg, Jon, and Eva Tardos. ***Algorithm design***. Pearson Education India, 2006.
2. Aho A.V., Hopcroft J.E., and Ullman J.D., ***Data Structures and Algorithms***, Pearson Education, New Delhi, 1983.
3. Sahni S., Data Structures, ***Algorithms, and Applications in C***++, Mc Graw Hill, Singapore, 1998.

Web Reference

1. **https://gcc.gnu.org/onlinedocs/gcc/Option-Summary.html**
2. **https://www.gnu.org/software/gdb/documentation/**
3. **https://ftp.gnu.org/old-gnu/Manuals/gprof-2.9.1/html_mono/gprof.html**

Course Contents and Lecture Schedule

Topic	No. of hours
1. Advanced use of gcc : Important Options -o, -c, -D, -l, -I, -g, -O, -save-temps, -pg	1
2. Familiarisation with gdb : Important Commands - break, run, next, print, display, help	1
3. Using gprof : Compile, Execute and Profile	1
4. Review of Basic Data Structures (Array, List, Stack, Queue,Trees) a. Merge two sorted arrays and store in a third array b. Circular Queue - Add, Delete, Search c. Singly Linked Stack - Push, Pop, Linear Search d. Doubly linked list - Insertion, Deletion, Search e. Binary Search Trees- Insertion, Deletion, Search	8
5. Set Data Structure and set operations (Union, Intersection and Difference) using Bit String.	3
6. Disjoint Sets and the associated operations (create, union, find)	3
7. Binomial Heaps and operations (Create, Insert, Delete, Extract-min, Decrease key)	4
8. B Trees and its operations	4
9. Red Black Trees and its operations	4
10. Graph Traversal techniques (DFS and BFS) and Topological Sorting	4
11. Finding the Strongly connected Components in a directed graph	3
12. Prim's Algorithm for finding the minimum cost spanning tree	3
13. Kruskal's algorithm using the Disjoint set data structure	3
14. Single Source shortest path algorithm using any heap structure that supports mergeable heap operations	3

SEMESTER 2

SEMESTER – 2

20MCA102	ADVANCED DATABASE MANAGEMENT YSTEMS	CATEGORY	L	T	P	CREDIT
		GENERAL	3	1	0	4

Preamble: This course provides the basic concepts and terminology related to relational and non-relational database management systems. The concept of advanced DBMS techniques and new generation databases like MongoDB, HBase and Cassandra are also introduced. This course serves as a prerequisite for many advanced courses in Data Science and Machine Learning areas.

Prerequisite: Basic knowledge in Database Management Systems.

Course Outcomes: After the completion of the course the student will be able to

CO 1	Understand the fundamentals of relational database systems including: data models, database architectures and ER features.
CO 2	Analyze and apply the different normalization techniques.
CO 3	Assess the basic issues of transaction processing and concurrency control.
CO 4	Understand the roles that databases play in organizations and familiarize with basic database storage, file organization, database accessing techniques.
CO 5	Understand the basics of query processing, object-oriented, distributed databases.
CO 6	Analyze non-relational database systems and structures and XML.

Mapping of course outcomes with program outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1				1			1					
CO 2	3	3	3	2			2	2			2	2
CO 3	1	2	2	2		2					2	2
CO 4					1		1					
CO 5	1			1								
CO 6	1											

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination
	1	2	
Remember(K1)	10	10	10
Understand(K2)	20	20	20
Apply(K3)	20	20	30
Analyse(K4)			
Evaluate(K5)			
Create(K6)			

Mark distribution

Total Marks	CIE	ESE	ESE Duration
100	40	60	3 hours

Continuous Internal Evaluation Pattern:

Attendance	: 8 marks
Continuous Assessment Test (2 numbers)	: 20 marks
Assignment/Quiz/Course project	: 12 marks

End Semester Examination Pattern: There will be two parts; Part A and Part B. Part A contains 10 compulsory short answer questions, 2 from each module. Each question carries 3 marks. Part B contains 2 questions from each module of which student should answer any one. Each question can have maximum 2 sub-divisions and carry 6 marks.

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Examine why databases are important. (K3)
2. Describe the basic features of the relational data model and discuss their importance to the end user and the designer. (K2)
3. Analyze the graphic depiction of relationships among the entities and examine how these depictions help in the database design process. (K3 & K4)

Course Outcome 2 (CO2):

1. Evaluate and design good table structures to control data redundancies and anomalies. (K5 & K6)

Course Outcome 3(CO3):

1. Explain the database transaction and its properties. (K2)
2. Describe concurrency control and analyze the role it plays in maintaining the database integrity. (K2 & K4)
3. Assess the common algorithms for concurrency control. (K5)
4. Define deadlock and discuss the strategies for managing deadlocks. (K1 & K2)
5. Examine how database recovery management is used to maintain database integrity. (K3)

Course Outcome 4 (CO4):

1. Discuss the various disk-organization techniques. (K2)
2. Describe the various data structures that allow fast access to data. (K2)
3. Analyze and examine the different indexing techniques. (K3 & K4)

Course Outcome 5 (CO5):

1. Describe the basics of query processing and evaluate the query processing cost. (K2 & K5)
2. Analyze the concept of object oriented databases and distributed databases. (K4)

Course Outcome 6 (CO6):

1. Explain the concept of XML. (K2)
2. Describe the various NoSQL databases. (K2)

Model Question Paper

Reg No.:________________		Name:____	
APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY MODEL QUESTION PAPER M.C.A.DEGREE EXAMINATION			
Course Code: 20MCA102			
Course Name: ADVANCED DATABASE MANAGEMENT SYSTEMS			
Max. Marks: 60		Duration: 3 Hours	
PART A			
		Answer all questions, each carries3 marks.	Marks
1		Define weak entity set with an example.	(3)
2		With the help of a diagram explain the different levels of data abstraction?	(3)
3		Differentiate between BCNF and 3NF with an example.	(3)
4		Explain functional dependency with suitable example.	(3)
5		Discuss the ACID properties of transaction.	(3)
6		Define deadlock and discuss the strategies for managing deadlocks.	(3)
7		Diagrammatically represent the basic steps in query processing.	(3)
8		Differentiate static and dynamic hashing.	(3)
9		Illustrate the different types of Distributed Databases.	(3)
10		Define collection and document in MongoDB.	(3)
PART B			
Answer any one question from each module. Each question carries 6 marks.			
Module I			
11		Draw an E-R diagram of a college database with entities student, staff, course, teacher, clerk, department & hostel? Relationship names must be meaningful	(6)

		and there should be an ISA relationship also in diagram.	
		OR	
12		Explain the relational model concept and discuss the different relational model constraints.	(6)
		Module II	
13		Analyse the common anomalies found in databases? How can we eliminate it through normalization?	(6)
		OR	
14		Define Normalization. Explain 1NF, 2NF and 3NF in detail.	(6)
		Module III	
15		Define deadlock. How can we deal with deadlocks?	(6)
		OR	
16		Explain concurrency control with locking methods.	(6)
		Module IV	
17		Explain the various RAID levels with appropriate diagrams.	(6)
		OR	
18		Differentiate between Dense index and Sparse index with example.	(6)
		Module V	
19		Explain HBase and Cassandra.	(6)
		OR	
20		Explain XML and its applications.	(6)
		****	

SYLLABUS

Module I:

Relational Databases:- Introduction - Purpose of Database System – Database System Applications - View of data: Data Abstraction, Instances and Schemas, Data Models – Database Architecture - Database Users and Administrators: Database Users and Interfaces, DBA – Introduction to the Relational Model: Structure of Relational Database, database Schema, Keys, Relational Query language – The Relational Algebra: Fundamental Operations, Formal definition of the relational algebra, additional relational algebra operations – The Entity-Relationship model: Entity Set, Relationship Set, Attributes – Constraints: Mapping cardinalities, Key Constraints, Participation Constraints - E-R Diagrams: Basic structure, Complex attributes, Roles, Non binary relationship sets, Weak Entity Set, Relational Database Design using ER- to Relational Mapping – Extended ER Features: Specialization, Generalization, Attribute inheritance, Constraints on generalization, Aggregation.

Module II:

Database Design:- Database Tables and Normalization – The Need for Normalization – The Normalization Process: Inference Rules for Functional Dependencies (proof not needed) - Minimal set of Functional Dependencies - Conversion to First Normal Form, Conversion to Second Normal Form, Conversion to Third Normal Form - Improving the Design - Surrogate Key Considerations - Higher Level Normal Forms: Boyce/Codd Normal Form, Fourth Normal Form, Join dependencies and Fifth Normal Form – Normalization and Database Design.

Module III:

Transaction Management and Concurrency Control:- Transaction: Evaluating Transaction Results, Transaction Properties, Transaction Management with SQL, The Transaction Log – Concurrency Control: Lost Updates, Uncommitted Data, Inconsistent Retrievals, The Scheduler – Concurrency Control with Locking Methods: Lock Granularity, Lock Types, Two Phase Locking to Ensure Serializability, Deadlocks – Concurrency Control with Timestamping Methods: Wait/Die and Wait/Wound Schemes – Concurrency Control with Optimistic Methods - Database Recovery Management: Transaction Recovery.

Module IV:

Data Storage and Querying:- RAID – File Organization – Organization of Records in Files – Indexing and Hashing: Basic concept, Ordered Indices, B+ tree Index Files: Structure of a B+-Tree (structure only, algorithms not needed) - B tree index files – Static Hashing – Dynamic Hashing – Query Processing: Overview - Selection Operation.

Module V:

System Architecture, Object Oriented Databases, XML and NoSQL:- Distributed Databases: Homogeneous and Heterogeneous Databases, Distributed Data Storage,

Distributed Transactions - Object Based Databases: Overview, Complex Data types, Structured types and inheritance in SQL, Table Inheritance, Array and Multiset types in SQL, Object identity and reference types in SQL - XML: DTD and XML Schema, XML presentation, XML Applications - Next Generation Databases: Distributed Relational Databases - Nonrelational Distributed Databases - MongoDB Sharding and Replication - Hbase - Cassandra - CAP Theorem.

Text Books

1. Abraham Silberschatz, Henry F. Korth, S. Sudarshan," ***Database System Concepts***", McGraw Hill Education, 6th Edition, 2011. *(for Module 1 Refer Chapter 1 [1.1 to 1.3, 1.9,1.12], Chapter 2 [2.1-2.3,2.5], Chapter 6 [6.1], Chapter 7 [7.2, 7.3, 7.8(7.81. To 7.8.5)], for Module 4 Refer Chapter 10 [10.3, 10.5, 10.6], Chapter 11 [11.1, 11.2, 11.3(11.3.1), 11.4.5 and module 5 Refer Chapter 19 [19.1,19.2, 19.3 - Distributed Databases], Refer Chapter 22 [22.1 to 22.6 - Object Based Databases]).*
2. Ramez Elmasri, Shamkant B.Navathe, " ***Fundamentals of Database Systems*** ", Pearson Education, 5th Edition, 2007. *(for Module 1 - Refer Chapter 7 [7.1] - 7.1.1 - Relational Database Design using ER- to Relational Mapping]) and for Module 2 - Refer Chapter 10 [10.2.2 and 10.2.4], Refer Chapter 11 [11.4 - Join dependencies and Fifth Normal Form).*
3. Guy Harrison, "***Next Generation Databases: NoSQL, NewSQL, and Big Data***", Apress, 1st Edition, 14 December 2015. *Refer Chapters 8 and 3 (for Module 5 - Next Generation Databases and CAP Theorem).*

4. Rob, Peter and Carlos Coronel, "***Database Principles: Fundamentals of Design, Implementation and Management***", 9th Edition, 2011. *(for Module 2, refer chapter 6) and (for module 3, refer chapter 10) and (for Module 5, refer Chapter 14 -XML).*

Reference Books

1. Ashutosh Kumar Dubay, "***Database Management Concepts***", S.K. Kataria & Sons, 1st Edition (2012).
2. Raghu Ramakrishnan and Johannes Gehrke, "***Database Management Systems***",

   McGraw Hill, 3rd Edition (2014).
3. Thomas M Connolly and Carolyn E Begg, "***Database systems- A Practical Approach to Design, Implementation and Management***", Pearson Education, 4th Edition (2014).

Web Resources

1. Introduction to Databases (nptel) https://nptel.ac.in/courses/106/106/106106220/
2. Database Design (nptel) https://nptel.ac.in/courses/106/106/106106093/
3. Introduction to Database Systems and Design
   https://nptel.ac.in/courses/106/106/106106095/
4. Fundamentals of Database Systems

   https://nptel.ac.in/courses/106/104/106104135/#
5. Database Management Essentials (Coursera)
   https://www.coursera.org/learn/database-management
6. Database Systems Concepts & Design

   https://www.udacity.com/course/database-systems-concepts-design--ud150

Course Contents and Lecture Schedule

No.	Topic	No. of Lectures
1	***Module I: Relational Databases***	***15 hrs***
1.1	Introduction - Purpose of Database System - Database System Applications	1 hr
1.2	View of data: Data Abstraction, Instances and Schemas, Data Models	1 hr
1.3	Database Architecture	1 hr
1.4	Database Users and Administrators: Database Users and Interfaces, DBA	1 hr
1.5	Introduction to the Relational Model: Structure of Relational Database, database Schema, Keys, Relational Query language	1 hr
1.6	The Relational Algebra: Fundamental Operations, Formal definition of the relational algebra, additional relational algebra operations	2 hr
1.7	The Entity-Relationship model: Entity Set, Relationship Set, Attributes	1 hr
1.8	Constraints: Mapping cardinalities, Key Constraints, Participation Constraints	2 hr
1.9	E-R Diagrams: Basic structure, Complex attributes, Roles, Non binary relationship sets, Weak Entity Set	1 hr
1.10	Relational Database Design using ER- to Relational Mapping	2 hr
1.11	Extended ER Features: Specialization,Generalization, Attribute inheritance, Constraints on generalization, Aggregation.	2 hr
2	***Module II: Database Design***	***9 hrs***
2.1	Database Tables and Normalization - The Need for Normalization	1 hr
2.2	The Normalization Process: Inference Rules for Functional Dependencies (proof not needed) - Minimal set of Functional Dependencies - Conversion to First Normal Form, Conversion to Second Normal Form	2 hr
2.3	Conversion to Third Normal Form	1 hr
2.4	Improving the Design - Surrogate Key Considerations	1 hr

No	Topic	No. of Lectures
2.5	Higher Level Normal Forms: Boyce/Codd Normal Form	1 hr
2.6	Fourth Normal Form	1 hr
2.7	Join dependencies and Fifth Normal Form	1 hr
2.8	Normalization and Database Design	1 hr
3	***Module III: Transaction Management and Concurrency Control***	***9 hrs***
3.1	Transaction: Evaluating Transaction Results, Transaction Properties	1 hr
3.2	Transaction Management with SQL, The Transaction Log	1 hr
3.3	Concurrency Control: Lost Updates, Uncommitted Data, Inconsistent Retrievals, The Scheduler	2 hr
3.4	Concurrency Control with Locking Methods: Lock Granularity	1 hr
3.5	Lock Types, Two Phase Locking to Ensure Serializability	1 hr
3.6	Deadlocks	1 hr
3.7	Concurrency Control with Timestamping Methods: Wait/Die and Wait/Wound Schemes, Concurrency Control with Optimistic Methods, Database Recovery Management: Transaction Recovery	2 hr
4	***Module IV: Data Storage and Querying***	***10 hrs***
4.1	RAID	1 hr
4.2	File Organization	1 hr
4.3	Organization of Records in Files	1 hr
4.4	Indexing and Hashing: Basic concept, Ordered Indices	1 hr
4.5	B+ tree Index Files: Structure of a B+-Tree, B tree Index Files	2 hr
4.6	Static Hashing, Dynamic Hashing	2 hr
4.7	Query Processing: Overview, Selection Operation	2 hr

No	Topic	No. of Lectures
5	***Module V: System Architecture, Object Oriented Databases, XML and NoSQL***	***13 hrs***
5.1	Distributed Databases: Homogeneous and Heterogeneous Databases, Distributed Data Storage, Distributed Transactions	2 hr
5.2	Object Based Databases: Overview, Complex Data types	1 hr
5.3	Structured types and inheritance in SQL	1 hr
5.4	Table Inheritance	1 hr
5.5	Array and Multiset types in SQL	1 hr
5.6	Object identity and reference types in SQL	1 hr
5.7	XML: DTD and XML Schema	1 hr
5.8	XML presentation, XML Applications	1 hr
5.9	Next Generation Databases: Distributed Relational Databases - CAP Theorem	1 hr
5.10	Norelational Databases – MongoDB Sharding and Replication	1 hr
5.11	Hbase	1 hr
5.12	Cassandra	1 hr

20MCA104	ADVANCED COMPUTER NETWORKS	CATEGORY	L	T	P	CREDIT
		GENERAL	3	1	0	4

Preamble: This course intends to provide insight into Advanced Computer Networks. A software professional should have an understanding of layered network architecture. Various kinds of network architectures, issues in integrating networks to modern application development are to be addressed. It is also intended to expose the student to modern technologies such as IPV6 and software defined networks. More detailed treatment can be done through seminars, assignments and talks by eminent external experts.

Prerequisite: Basic concepts of computer operating systems.

Course Outcomes: After the completion of the course the student will be able to

CO 1	Comprehend the terminology and concepts of basic communication model, analyse the protocol layers and design application layer protocols.
CO 2	Understand and analyse the various transport layer protocols.
CO 3	Compare and contrast various routing algorithms in the network layer.
CO 4	Understand and analyse the concepts of link layer and physical layer.
CO 5	Understand how modern cellular and wireless networks work

Mapping of course outcomes with program outcomes

	PO 1	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO 10**	**PO 11**	**PO 12**
CO 1	3	3	2		2	2	2		3		2	
CO 2	3	3	2		2	2			3		2	
CO 3	3	3			2	2	2		3		2	
CO 4	3	3				2			3		2	
CO 5	3	3				2			3			

Assessment Pattern

Bloom's Category	**Continuous Assessment Tests**		**End Semester Examination**
	1	**2**	
Remember	10	10	10
Understand	20	20	20
Apply	20	20	30
Analyze			
Evaluate			
Create			

Mark distribution

Total Marks	CIE	ESE	ESE Duration
100	40	60	3 hours

Continuous Internal Evaluation Pattern:

Attendance	: 8 marks
Continuous Assessment Test (2 numbers)	: 20 marks
Assignment/Quiz/Course project	: 12 marks

End Semester Examination Pattern: There will be two parts; Part A and Part B. Part A contains 10 compulsory short answer questions, 2 from each module. Each question carries 3 marks. Part B contains 2 questions from each module of which student should answer any one. Each question can have maximum 2 sub-divisions and carry 6 marks

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Explain HTTP request-response behavior with a neat diagram.
2. Compare and contrast OSI and TCP/IP network reference models.
3. Explain the importance of layering in data communication.

Course Outcome 2 (CO2)

1. Explain the process of three-way handshaking in TCP.
2. Compare and contrast Multiplexing and De-multiplexing process in transport layer.
3. Explain How TCP is controlling congestion during data transmission.

Course Outcome 3(CO3):

1. Explain how multicast routing is used in routing protocols.
2. Compare and contrast IPV4 and IPV6.
3. Differentiate virtual circuits and datagram networks.

Course Outcome 4 (CO4):

1. Explain how parity is used to achieve error detection in data communication.
2. Illustrate IEEE 802.3 frame structure.
3. Write short notes on routers, switches and bridges.

Course Outcome 5 (CO5):

1. List out and explain the various IEEE 802.11 WLAN Components.
2. Explain the architecture of Bluetooth in personal area networks.
3. Explain any six network attacks and their counter measures.

Model Question paper

Part A

1. Differentiate HTTP persistent and non-persistent communication.
2. List out and explain the functionalities of different DNS records.
3. Compare TCP and UDP at transport layer.
4. Demonstrate how stop-and-wait protocol is used for reliable data transfer.
5. Explain how IPv6 solve the problem of IPv4 exhaustion?
6. Explain how ARP is working in data link layer?
7. A series of 8-bit message blocks to be transmitted across a data link using CRC for error detection. A generator polynomial of x^3+ x^2+1.is to be used. Message transmitted as 110010. Explain how CRC check is implemented?
8. Classify various wired media used in short and long distance communication.
9. Explain Network Address Translation (NAT).
10. Explain piconet topology of Bluetooth? [3x10=30 Marks]

Part B

Module 1

11. List and explain ISO/OSI layers and their functions. [6 Marks]

OR

12. Describe various service models in Quality of Service (QOS). [6 Marks]

Module 2

13. Write a short note on:
    a. Stop-and-wait [3 Marks]
    b. Go-back-N [3 Marks]

OR

14. Explain the principles of congestion control with its fairness and efficiency. [6 Marks]

Module 3

15. Define routing? Explain the process of link state routing with OSPF protocol.

OR [6 Marks]

16. What is Virtual circuit? Explain the connection management in Virtual circuit with suitable diagrams.

[6 Marks]

Module 4

17. Write a short note on:

a. Collision based multiple access protocol [3 Marks]
b. Token based multiple access protocol [3 Marks]

OR

18. Explain IEEE 802.3 Ethernet frame format with its access protocol.

[6 Marks]

Module 5

19. What is Bluetooth? Explain the various layers of Bluetooth with a neat diagram.
[6 Marks]

OR

20. Write a short note on:
a. Traffic analysis tools [3 Marks]
b. Troubleshooting [3 Marks]

Syllabus

Module	Contents	Hours
I	Overview of Computer Networks and the Internet. History. Protocols, Review of last mile technologies used for internet access. Packet switching. Basic ideas about delay queuing throughput. Concept of Quality of Service, Protocol layering . OSI model and TCP model Application layer protocols - Client-server architecture Network layer 7 application architecture, Web, HTTP, FTP, SMTP, POP3, and DNS, Peer-to-peer file sharing networks	10

Module	Contents	Hours
II	Transport Layer Protocols: Introduction to transport layer, Multiplexing and de-multiplexing, Principles of Reliable data transfer - Stop-and-wait and Go-back- N design and evaluation, Connection oriented transport TCP, Connectionless transport UDP, Principles of congestion control -efficiency and fairness	10
III	Network Layer Protocols: Virtual circuits and datagrams, Principles of routing, internet protocol Ipv4 CIDR Routing algorithms: Link-state and distance vector routing, Routing on the internet RIP OSPF and BGP, Multicast routing. Introduction to IPV6 and software defined networks, Open flow	10
IV	Link layer and Physical Layer: Introduction to link layer - Error detection (parity, checksum, and CRC), Multiple access protocols (collision and token based), IEEE 802.3 Ethernet, Switching and bridging, Media, Signal strength and interference. Data encoding. Ethernet switches , Routers MAC, ARP, FIB	8
V	**I**EEE 802.11 Wi-Fi, Bluetooth, and cellular networks,Threats and attacks, Network Address Translation , Firewalls, VPNs, Introduction to network management, SNMP, Overview of tools and troubleshooting, Traffic analysis tools and Configuration management.	10

Textbooks:

1. Behrouz A Forouzan, Firouz Mosharraf, "***Computer Networks: A top down Approach***", McGraw Hill Education, 1 st Edition (2011).
2. James F Kurose and Keith W Ross, "***Computer Networking: A Top - Down Approach***", Pearson Education; 6 th Edition (2017).

Reference Books:

1. Kevin R. Fall, W. Richard Stevens, "***TCP/IP Illustrated, Volume 1 -The Protocols***", Pearson Education, 2 nd Edition (2014).
2. Larry Peterson, Bruce Davie, "***Computer Networks, A systems Approach***", Morgan Kaufmann Publishers, 5th Edition (2011).
3. Uyless Black, "***Computer Networks: Protocols, Standards and Interface***", Prentice HallIndia Learning Private Limited, 8 th Edition (2015).
4. William Stallings, "***Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud***", Pearson Education, 1 st Edition (2016)
5. ***The Illustrated Network: How TCP/IP Works in a Modern Network*** 2nd edition Walter Goralski Morgan Kaufmann Publications

Course Contents and Lecture Schedule

No	Topic	No. of Lectures
1		
1.1	Overview of Computer Networks and the Internet. History. Protocols , Review of last mile technologies used for internet access. Packet switching.	2
1.2	Basic ideas about delay queuing through put. Concept of Quality of Service Protocol layering . OSI model and TCP model	4
1.3	Application layer protocols - Client-server architecture Network application architecture, Web, HTTP, FTP, SMTPPOP3 and DNS, Peer-to-peer file sharing networks	4
2		
2.1	Transport Layer Protocols: Introduction to transport layer	2
2.2	Multiplexing and demultiplexing, Principles of Reliable data transfer - Stop-and-wait and Go-back- N design and evaluation	3
2.3	Connection oriented transport TCP, Connection less transport UDP	3
2.4	Principles of congestion control -efficiency and fairness	2
3		
3.1	Network Layer Protocols: Virtual circuits and datagrams	2
3.2	Principles of routing, internet protocol Ipv4 NAT , Routing algorithms: Link-state and distance vector routing,	3
3.3	Routing on the internet RIP OSPF and BGP, Multicastrouting.	2
3.4	Introduction to IPV6 and software defined networks	2
4		
4.1	Link layer and Physical Layer: Introduction to link layer - Error detection (parity, checksum, and CRC)	2
4.2	Multiple access protocols (collision and token based), IEEE 802.3	2
4.3	Ethernet, Switching and bridging, Media, Signal strength and interference. Data encoding. Ethernet switches , Routers MAC, ARP, FIB	4

No	Topic	No. of Lectures
5		
5.1	**I**EEE 802.11 Wi-Fi, Bluetooth, and cellular networks,	3
5.2	Threats and attacks, Firewalls, NAT,VPNs, Introduction to network management, SNMP,	4
5.3	Overview of tools and troubleshooting, Traffic analysis tools and Configuration management**.**	3

20MCA162	APPLIED STATISTICS	CATEGORY	L	T	P	CREDIT
		ELECTIVE	3	1	0	4

Preamble: This course introduces the concepts and application of probability distribution, Correlation, Regression and testing of hypothesis. The topics treated in this course have applications in Computer Science.

Course Outcomes: After the completion of the course the student will be able to

CO 1	Apply the concept of discrete probability distributions in determining the parameters of the distribution and hence to solve different problems
CO 2	Apply the concept of continuous probability distribution in solving different problems
CO 3	Apply the principles of correlation and regression in practical problems.
CO 4	Develop confidence intervals for various problems.
CO 5	Test the given hypothesis on the basis of known criteria.

Mapping of course outcomes with program outcomes

	PO 1	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO 10**	**PO 11**	**PO 12**
CO 1	3	3										
CO 2	3	3										
CO 3	3	3										
CO 4	3	3										
CO 5	3	3										

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination
	1	2	
Remember(K1)	10	10	10
Understand(K2)	20	20	20
Apply(K3)	20	20	30
Analyse(K4)			
Evaluate(K5)			
Create(K6)			

Mark distribution

Total Marks	CIE	ESE	ESE Duration
100	40	60	3 hours

Continuous Internal Evaluation Pattern:

Attendance	: 8 marks
Continuous Assessment Test (2 numbers)	: 20 marks
Assignment/Quiz/Course project	: 12 marks

End Semester Examination Pattern: There will be two parts; Part A and Part B. Part A contains 10 compulsory short answer questions, 2 from each module. Each question caries 3 marks. Part B contains 2 questions from each module of which student should answer any one.Each question can have maximum 2 sub-divisions and carry 6 marks

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. With the usual notation find p for the binomial random variable X, if n = 6 and

9 p[x =4] = p[x =2] (K3)

2. Define Poisson distribution. Derive it's Mean. (K1)

3. A die is tossed twice. Getting 'a number greater than 4' is considered as success. Find the mean and variance of the probability distribution of the number of success. (K3)

Course Outcome 2 (CO2)

1. Define distribution function of a continuous random variable. Also state it’s important properties. (K1)
2. Derive the mean and variance of a continuous uniform distribution. (K4)
3. In a normal distribution 31% of the items are under 45 and 8% are over 64. Find the mean and standard deviation of the distribution. (K3)

Course Outcome 3(CO3):

1. State the principle of least squares. (K1)
2. Fit a parabola by the method of least squares, to the following data. (K3)

x:	1	2	3	4	5
y:	5	12	26	60	97

3. Compute the correlation coefficient from the following data. (K3)

x:	77	54	27	52	14	35	90	25	96	60
y:	35	58	60	40	50	40	35	56	34	42

Course Outcome 4 (CO4):

1. Differentiate parameter and statistic. (K1)
2. A random sample of 700 units from a large consignment showed that 200 were damaged. Find i) 95% and ii) 99% confidence limits for proportion of damaged limits in the consignment. (K3)
3. Explain different types of sampling. (K2)

Course Outcome 5 (CO5):

1. State Type I and Type II error. (K1)
2. Explain the different steps in testing of hypothesis. (K2)
3. In a big city 325 men out of 600 men were found to be smokers. Does this information support the conclusion that the majority of men in this city are smokers? (K5)

Syllabus

Module 1

Introduction – Random Experiment, Random Variables, Discrete Random Variables, Probability Distributions and Probability Mass Functions, Mean and Variance of a Discrete Random Variable, Discrete Uniform Distribution - Mean and Variance, Binomial Distribution - Mean and Variance, Geometric Distribution - Mean and Variance, Poisson Distribution - Mean and Variance

Module 2

Continuous Random Variables, Probability Density Functions, Mean and Variance of a Continuous Random Variable, Continuous Uniform Distribution- Mean and Variance, Normal Distribution-Mean and Variance (Proof not required), Standard Normal Distribution, Exponential Distribution.

Module 3

Curve fitting – Principle of least squares – fitting a straight line – fitting a parabola – linear correlation and regression – Karl's Pearson's Coefficient of Correlation.

Joint and marginal probability distribution - Conditional probability distribution - independent random variable (discrete case only).

Module 4

Sampling distribution – Introduction to sampling – random sampling – sampling distribution – standard error – estimation – interval estimates and confidence interval – estimation of population mean and proportions (small and large samples).

Module 5

Testing of hypothesis – introduction – basic concepts – Hypothesis concerning a mean – equality of means – Hypothesis concerning one proportion – difference of two proportions.

Text Books

1. Veerarajan T, "***Probability and Random Process***", 3rd Edition, Tata McGraw-Hill(2002)

2. Gupta S.C and Kapoor V .K, "***Fundamentals of Mathematical Statistics***", Sultan Chand and Sons (2014).

Reference Books

1. David S. Moore and George P. McCabe, "***Introduction to practice of statistics***", W.H. Freeman & Company, 5th Edition (2005).

2. G. Jay Kerns, "***Introduction to Probability and Statistics Using R***", Chapman & Hall (2010)
3. Douglas C. Montgomery and George C. Runger, "***Applied Statistics and Probability for Engineers***", Wiley India, 5th Edition (2012).

Web Resources

1. Probability and statistics EBook
http://wiki.stat.ucla.edu/socr/index.php/EBook
2. https://www.openintro.org/stat/textbook.php
3. http://www.math.uah.edu/stat/index.html
4. Statistics Online Computational Resource
http://www.socr.ucla.edu/

Course Contents and Lecture Schedule

Topic	No. of lectures
Module 1	**9 hrs**
Introduction – Random Experiment, Random Variables, Discrete Random Variables, Probability Distributions and Probability Mass Functions, Mean and Variance of a Discrete Random Variable	3
Discrete Uniform Distribution - Mean and Variance	1
Binomial Distribution - Mean and Variance	2
Geometric Distribution - Mean and Variance, Poisson Distribution - Mean and Variance	3
Module 2	**9 hrs**
Continuous Random Variables, Probability Density Functions, Mean and Variance of a Continuous Random Variable	3
Continuous Uniform Distribution, Mean and Variance	2
Normal Distribution, Mean and Variance (Proof not required), Standard Normal Distribution	3
Exponential Distribution	1

Topic	No. of lectures
Module 3	**9 hrs**
Curve fitting – Principle of least squares – fitting a straight line – fitting a parabola	3
linear correlation and regression – Karl’s Pearson’s Coefficient of Correlation	2
Joint and marginal probability distribution	2
Conditional probability distribution - independent random variable (discrete case only)	2
Module 4	**9 hrs**
Sampling distribution – Introduction to sampling – random sampling	3
sampling distribution – standard error	2
estimation – interval estimates and confidence interval – estimation of population mean and proportions (small and large samples)	4
Module 5	**9 hrs**
Testing of hypothesis – introduction – basic concepts	3
Hypothesis concerning a mean – equality of means	3
Hypothesis concerning one proportion – difference of two proportions	3

20MCA164	ORGANIZATIONAL BEHAVIOUR	CATEGORY	L	T	P	CREDIT
		ELECTIVE	3	1	0	4

Preamble: This course is designed primarily for students who are being exposed to Organizational Behaviour for the first time. Primary aim of this course is to help students to understand the organizational culture and its dynamics and to acquire skills to take rational decisions in groups or organizations.

Prerequisite: Nil

Course Outcomes: After the completion of the course the student will be able to

CO 1	Identify managers’ challenges and opportunities in applying OB concepts.
CO 2	Analyse various characteristics of individual behaviour and its impact on organizational performance.
CO 3	Acquire knowledge about the complexities associated with management of individual behaviour in the organization.
CO 4	Understand group behaviour and develop inter-personal skills and group dynamics.
CO 5	Understand organizational structures and analyze the behavioral implications of different organizational designs.

Mapping of course outcomes with program outcomes

	PO 1	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO 10**	**PO 11**	**PO 12**
CO 1	2	2	1			2		2	2		1	1
CO 2	2	2				1	2	3	3		3	1
CO 3	2	2				1	2	3	3		3	1
CO 4	2	2				1		3	3		3	
CO 5	2	2	1			2		2	2		1	

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination
	1	2	
Remember(K1)	30%	30%	30%
Understand(K2)	30%	30%	30%
Apply(K3)			
Analyse(K4)	30%	30%	30%
Evaluate(K5)	10%	10%	10%
Create(K6)			

Mark distribution

Total Marks	CIE	ESE	ESE Duration
100	40	60	3 hours

Continuous Internal Evaluation Pattern:

Attendance	: 8 marks
Continuous Assessment Test (2 numbers)	: 20 marks
Assignment/Seminar/Course project	: 12 marks

End Semester Examination Pattern: There will be two parts; Part A and Part B. Part A contains 10 compulsory short answer questions, 2 from each module. Each question caries 3 marks. Part B contains 2 questions from each module of which student should answer any one. Each question can have maximum 2 sub-divisions and carry 6 marks

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Describe the importance of inter-personal skills in the workplace.
2. Analyse the challenges and opportunities for managers in using OB concepts.
3. "The workplace discriminations undermine organisational performance", Justify.

Course Outcome 2 (CO2):

1. Identify the major job attitude and job satisfaction parameters.
2. How to apply concepts about emotions and moods to specific OB issues.
3. Differentiate between person fit for job and person fit for organisation.

Course Outcome 3(CO3):

1. What is learning and what are the theories of learning?
2. How do individual differences and organisational constraints influence decision making?
3. Identify how employee involvement measures motivate employees.

Course Outcome 4 (CO4):

1. Differentiate group and team.
2. Relate the contemporary theories of leadership to earlier foundational theories.
3. What are three types of conflicts and the three loci of conflict?

Course Outcome 5 (CO5):

1. What are the functional and dysfunctional effects of organisational culture?
2. What are your suggestions to overcome resistance to change in an organization?
3. Identify the potential environmental, organisational and personal sources of stress at work.

Model Question paper

Reg No.:________________		Name:_____________________________________ _	
APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY FIRST SEMESTER M.C.A.DEGREE EXAMINATION, MODEL QUESTION PAPER			
20MCA164 – Organisational Behaviour			
Max. Marks: 60			Duration: 3 Hours
PART A			
	Answer all questions, each carries 3 marks.		Marks
1		Define organisational behaviour. What is the objective of learning this subject in this programme?	3

2		What is workforce diversity? How to manage diversity?	3
3		State and explain the foundations of individual behaviour.	3
4		Differentiate ‘Classical conditioning’ and ‘Operant conditioning’ behavioural theories.	3
5		Describe how an understanding of attitudes is useful for the study of organisational behaviour.	3
6		What is job design? Describe different approaches to job design.	3
7		What is a team? Can groups become team? Defend your answer.	3
8		What is the difference between transformational leadership, transactional leadership and charismatic leadership?	3
9		What is creativity? How creativity can be enhanced in organisations?	3
10		What is Organisational Development? Why is it undertaken by organisations?	3
		PART B	
		Answer any one question from each module. Each question carries 6 marks.	
		Module I	
11		State your views on the following statement: “People influence organizations and organizations influence people”.	6
		OR	
12		Why have career management and talent management become important these days? Justify your points.	6
		Module II	
13		What is personality? What are its determinants? As per your opinion, which of them are more important in shaping personality.	6
		OR	
14		From your own experience, provide three examples of perceptual errors. Discuss the outcomes of each instance.	6

Module III			
15		Compare and contrast Maslow’s need hierarchy theory with Herzberg’s two-factory of motivation.	6
OR			
16		What is the relationship between stress and personality? What aspects of personality might tend to increase or decrease stress?	6
Module IV			
17		Why groupthink is to be avoided? How might a manager attempt to ensure that groupthink does not occur in his / her group?	6
OR			
18		What are the potential problems with upward and downward communications? How can managers alleviate these problems?	6
Module V			
19		What are the obstacles to change organisational culture? How can change be brought about?	6
OR			
20		What are the forces leading to change in organisations? Using Lewin’s theory justify why the change is resisted.	6
****			

Syllabus

Module 1

Nature of Organisational Behaviour: What are Organisations? – Why do Organisations Exist? – Nature of Organisational Behaviour – Foundations of OB – Contemporary OB – Scope of Organisational Behaviour – Contextual Perspectives of OB – Evolution of OB – OB Model

Management and Managers: Functions of Management – Manager’s Roles – Types of Managers –Evolution of Management Theory – Contemporary Trends in Management Thinking

Challenges in OB: Managing Inclusivity / Diversity – Career Management – Talent Management – Globalisation

Module 2

Foundations of Individual Behaviour: Personal Factors – Environmental Factors – Organisational Systems and Resources – Models of Individual Behaviour

Intelligence: Nature of Intelligence – Types of Intelligence – Model, Theories, Measurement of Intelligence – Factors Influencing Intelligence

Personality: Nature of Personality – The Shaping of Personality – Determinants of Personality – Personality Structure – OB Related Personality Traits

Perception and Attribution: Perception: Meaning and Definition – Factors Influencing Perception – Perceptual Process – Perception and OB

Learning: Explicit and Tacit Knowledge – How Learning Occurs? – Principles of Learning – Learning and OB

Module 3

Attitudes and Values: Nature of Attitudes – Components of Attitudes – Formation of Attitudes – Functions of Attitudes – Changing Attitudes – Work-related Attitudes – Values

Motivation: Nature of Motivation – Importance of Motivation – Motivational Challenges – Theories on Motivation

Applied Motivational Practices: Rewards – Job Design – Behaviour Modification – Empowerment –Problem Employees – Quality of Work Life – Employee Engagement

Work Stress: Work Stress Model – Burnout – Stress Management – Stress and Performance

Module 4

Group Dynamics: Nature of Groups – Types of Groups – Group Development – Usefulness & Pitfalls of Groups – Determinants of Group Behaviour – Group Structuring – Group Decision Making

Team Dynamics: Teams vs. Groups – Benefits from Teams – Types of Teams – Implementing Teams in Organisations – Team Properties – Effective Teamwork

Workplace Behaviour: Nature of Conflict – Changing Views of Conflict – Functional and Dysfunctional Conflict – The Process of Conflict – Levels of Conflict – Conflict Resolution – Conflict Management Styles - Managerial Implications – Negotiation and Conflict Resolution

Leadership: Nature of Leadership – Leadership and Management – Importance of Leadership – Formal and Informal Leadership – Leadership Styles and Their Implications – Theories of Leadership – Contemporary Issues on Leadership – Leadership Development

Communication: Interpersonal Communication – Organisational Communication – Communication Networks – Communication Roles – Informal Communication – Communication Media – Information Technologies – Managerial Implications

Module 5

Organisations: Nature of Organisations – Organisational Structure – Key Factors of Organisational Structure – Types of Organisational Structures – Organisations for Future – Informal Organisations – Managerial Implications

Organisational Culture: Cultural Dimensions – How is Culture Created? – Sustaining Culture – Effects of Culture – Changing Organisational Culture – Creativity in Organisations – Innovation in Organisations

Organisational Change and Development: Nature of Change – Levels of Change – Types of Change – Forces for Change in Organisations – Resistance to Change – Force Field Theory of Change - The Change Process – Organisational Development – Managerial Implications

Text Books

1. K Aswathappa, ***Organizational Behaviour***, Himalaya Publishing House, 2018.
2. Robbins, Stephen, Timothy, A & Sanghi, S. "***Organizational Behavior***",13th Edn, Pearson Education. 2009.

Reference Books

1. Mc Shane & Von Glinow, "***Organizational Behavior***", Mc Graw Hill Publications, New Delhi, 2008
2. ***Understanding Organizational Behaviour*** by Udai Pareek, Oxford University Press (Third Edition)
3. ***Behaviour in Organizations*** by Jerald Greenberg and Robert A. Baron, PHI learning private Ltd, New Delhi (Ninth Edition).
4. Laurie J. Mullins, ***Management and Organisational Behaviour***, Oxford Publishers, New Delhi, 2007.
5. ***ORGB*** by Nelson, Quick and Khandelwal, Cengage Learning New Delhi (second edition).

Course Contents and Lecture Schedule

Topic	No. of lectures (49 Hrs)
Module 1	**9 Hrs**
Nature of Organisational Behaviour	3
Management and Managers	3
Challenges in OB	3
Module 2	**10 Hrs**
Foundations of Individual Behaviour	2
Intelligence	2
Personality	2
Perception and Attribution	2
Learning	2
Module 3	**9 Hrs**
Attitudes and Values	2
Motivation	2
Applied Motivational Practices	3
Work Stress	2
Module 4	**12 Hrs**
Group Dynamics	2
Team Dynamics	2
Workplace Behaviour	3
Leadership	3
Communication	2
Module 5	**9 Hrs**
Organisations	3
Organisational Culture	3
Organisational Change and Development	3

20MCA166	FUNCTIONAL PROGRAMMING	CATEGORY	L	T	P	CREDIT
		ELECTIVE	3	1	0	4

Preamble: This course introduces a functional programming approach in problem solving. Salient features of functional programming like recursion, pattern matching, higher order functions are discussed.

Lists and their features, new types such as Recursive types, Enumerated types, Composite and Abstract types along with their applications are being discussed with high importance.

Haskell is introduced to give a practical flavour to the course.

Prerequisite: Discrete mathematics.

Course Outcomes: After the completion of the course the student will be able to

CO 1	Understand the principles of functional programming (Module 1)
CO 2	Write purely functional programs, using recursion, pattern matching, and higher- order functions ((Module 2)
CO 3	Design immutable data structures like lists. (Module 3)
CO 4	Understand generic types for functional programs (Module 4)
CO 5	Write programs using Haskell (Module 5)

Mapping of course outcomes with program outcomes

	PO 1	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO 10**	**PO 11**	**PO 12**
CO 1	2	2	2			2			2			
CO 2	2	2	2			2			2			
CO 3	2	2	2			2			2			
CO 4	2	2	2			2			2			
CO 5	2	2	2		2	2	2		2			2

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination
	1	2	
Remember	10	10	10
Understand	20	20	20
Apply	20	20	30
Analyse			
Evaluate			
Create			

Mark distribution

Total Marks	CIE	ESE	ESE Duration
100	40	60	3 hours

Continuous Internal Evaluation Pattern:

Attendance	: 8 marks
Continuous Assessment Test (2 numbers)	: 20 marks
Assignment/Quiz/Course project	: 12 marks

End Semester Examination Pattern: There will be two parts; Part A and Part B. Part A contains 10 compulsory short answer questions, 2 from each module. Each question carries 3 marks. Part B contains 2 questions from each module of which student should answer any one. Each question can have maximum 2 sub-divisions and carry 6 marks.

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Design a recursive function to add two numbers.
2. Design a tail recursive function to find the nth Fibonacci number.
3. Explain the basic differences between imperative style programming and functional style programming.

4. Analyse each of the following lambda expressions to clarify its structure. If the expression is a function, identify the bound variable and the body expression, and then analyse the body expression. If the expression is an application, identify the function and argument expressions, and then analyse the function and argument expressions:

i) λa.(a λb.(b a))

ii) λx.λy.λz.((z x) (z y))

iii) (λf.λg.(λh.(g h) f) λp.λq.p)

iv) λfee.λfi.λfo.λfum.(fum (fo (fi fee))

v) (λp.(λq.p λx.(x p)) λi.λj.(j i))

Course Outcome 2 (CO2)

1. Explain with the help of examples the various forms of function definitions.
2. Explain functional composition with the help of examples.
3. Deduce the type of the following expression:

(.) f g x = f (g x) where . -> Functional Composition.

Course Outcome 3(CO3):

1. Predict the output of the following along with detailed explanation on how did you arrive at the answer:
   A. [(a,b) | a <- [1 . . 8] ; even a; b <- [a + 3. . 4] ; odd b]
   B. ["Party" | k <- [1 .. 5]]
   C. [' * ' | i <- [1 .. 3] ; j <- [1, 2]]

2. Explain any three list operations along with function definitions and examples.

Note: Questions can be asked to solve problems using list comprehensions, to prove properties on list operations and functions on natural numbers using Mathematical Induction.

Course Outcome 4 (CO4):

1. Define Natural numbers as a Recursive Type and explain how this definition enumerates all Natural numbers.

2. Find the equivalent decimal representation of this value:

Succ (Succ (Succ (Succ (Succ (Succ Zero)))))

3. Define Fibonacci numbers using Pattern matching. Natural numbers should be represented as a Recursive type.

Note: Questions can be asked to prove properties on Binary Trees and Binary Search Trees using Structural Induction (Variant of Mathematical Induction.

Course Outcome 5 (CO5):

1. Duplicate only even numbers among the elements of a list using a Haskell function and explain. You need to do this in two ways; 1. Recursion 2. List Comprehension

Example : λ> dupli [1, 2, 3] ANS: [2,2]

Model question Paper

Duration : 3 Hours **PART A** **Total : 60 Marks**

1. Design a recursive function to add two numbers.
2. Can Arrays be used as a data structure in functional programming? Explain with reasons.
3. Explain functional composition with the help of an example.
4. Deduce the type of the following expression:

   (.) f g x = f (g x) where . -> Functional Composition.

5. Predict the output of the following along with detailed explanation on how did you arrive at the answer:
   a. [(a,b) | a <- [1 . . 8] ; even a; b <- [a + 3. . 4] ; odd b]
   b. ["Party" | k <- [1 .. 5]]
   c. [' * ' | i <- [1 .. 3] ; j <- [1, 2]]
6. Define the function “take”. For example, take does the following:

   Prelude> take 2 [1,2,3,4] [1,2].

7. Find the equivalent decimal representation of this value:

   Succ (Succ (Succ (Succ (Succ (Succ Zero)))))

8. Explain composite types with the help of an example.
9. Define a Haskell function to find the factorial of a given number.
10. Define a Haskell function to reverse the elements of a list.

(3 x 10 = 30 Marks)

PART B

11. Explain commonly used data types in functional programming along with their properties.

OR

12. Write Recursive definitions along with an explanation for the below Arithmetic operations. Illustrate the recursive flow with the help of a diagram.

1. add x y 2. mult x y 3. div x y

13. Explain the following along with suitable examples:

1. Currying in Functional Programming. 2. Strict and Non strict functions.

OR

14. Explain the following along with suitable examples:

1. Guards and Pattern matching.2. Inverse functions.

15. Given below the definition of a function funcky

```
funcky:: Int ->Int

funcky n

|n == 0 = 0

|otherwise = 1 + funcky ( n−1)
```

Predict the output of f for all n >= 0? Prove your answer.

OR

16. Explain any three list operations along with function definitions and examples.

17. Explain Recursive Data Types with the help of an example.

OR

18. Give the type definition of a binary tree along with explanation of two functions on binary trees.

19. Duplicate only even numbers among the elements of a list using a Haskell function and explain. You need to do this in two ways; 1. Recursion 2. List Comprehension

    Example : λ> dupli [1, 2, 3] ANS: [2,2]

OR

20. Define a queue data type in Haskell along with any two operations on it as well as examples.

(6 x 5 = 30 Marks)

SYLLABUS

Module I:

Review of recursion -Tail recursion -recursive program design- Functional Programming: Introduction, λ calculus, λ expressions, Identity function, Self application function, Function application function, Notation for naming functions and application reduction, Functions from functions, Argument selection and argument pairing functions,Free and bound variables,Name clashes and α conversion, Simplification through eta reduction, Conditions, Booleans and Integers, Recursion and Arithmetic, Expressions and values, Basic Data Types , Names and values in programming- Data structures in functional languages - Names and values in imperative and functional languages- Execution order in imperative and functional languages- Repetition in imperative and functional languages- Functions as values.

(Note : Recursion is a very important technique in functional programming, hence high importance needs to be given to make students understand the essentials of recursive thinking and program design, Basic Lambda (λ) calculus needs to be taught.)

Module II:

Functions: Functions and definitions, Functional composition, Operators, Inverse functions, Strict and non-strict functions, Type Inference.

(Note : Basic ways of defining functions, how to infer the types of variables and function needs to be taught)

Module III:

Lists: List notation, List comprehensions, Operations on lists, Map and filter, List patterns, Recursion and Induction: Over natural numbers, Over lists. Operations on lists

(Note : Mathematical Induction based Proofs needs to be taught from the reference text book.)

Module IV:

New Types : Enumerated types , Composite types , Recursive types , Abstract types , Trees: Binary trees , Binary search trees

(Note : Various definitions of properties of these new types, their property proofs etc needs to be taught)

Module V:

Programming with Haskell: Introduction to Haskell, Defining functions: guards, pattern matching and recursion, Lists, strings and tuples, Types and polymorphism, Higher order functions on lists: map, filter, list comprehension, User defined data types:lists, queues, trees

(Note : Students need to be taught how to program using Haskell in this module)

Text Books

1. Richard S. Bird, Philip Wadler, “Introduction to Functional Programming”,Prentice Hall , 1988 (Module 1,2,3,4)
2. Greg Michaelson, “An introduction to functional programming through lambda calculus”, Dover Publications, 2011 (Module 1)
3. Miran Lipovača, “Learn You a Haskell for Great Good!: A Beginner's Guide”, No Starch Press, 1st Edition (15 March 2011) (Module 5)

Reference Books and Sites

1. Simon Peyton Jones , “The Implementation of Functional Languages” , Prentice Hall.
2. Benjamin C. Pierce, " Types and Programming Languages", MIT Press, 2002
3. https://www.haskell.org/
4. http://learnyouahaskell.com

Course Contents and Lecture Schedule

No	Topic/Module	No. of Lectures
1	Review of recursion -Tail recursion -recursive program design-Functional Programming: Introduction, λ calculus, λ expressions, Identity function, Self application function, Function application function, Notation for naming functions and application reduction, Functions from functions, Argument selection and argument pairing functions,Free and bound variables,Name clashes and α conversion, Simplification through eta reduction, Conditions, Booleans and Integers, Recursion and Arithmetic, Expressions and values, Basic Data Types , Names and values in programming-Data structures in functional languages - Names and values in imperative and functional languages- Execution order in imperative and functional languages- Repetition in imperative and functional languages- Functions as values.	10
2	Functions: Functions and definitions, Functional composition, Operators, Inverse functions, Strict and non-strict functions, Type Inference.	8
3	Lists: List notation, List comprehensions, Operations on lists, Map and filter, List patterns, Recursion and Induction: Over natural numbers, Over lists. Operations on lists	10
4	New Types : Enumerated types , Composite types , Recursive types , Abstract types , Trees: Binary trees , Binary search trees	10
5	Programming with Haskell: Introduction to Haskell, Defining functions: guards, pattern matching and recursion, Lists, strings and tuples, Types and polymorphism, Higher order functions on lists: map, filter, list comprehension, User defined data types:lists, queues, trees	10

20MCA168	VIRTUALISATION AND CONTAINERS	CATEGORY	L	T	P	CREDIT
		ELECTIVE	3	1	0	4

Preamble:

A graduate course in Computer Applications should give due exposure to the recent developments. Since virtualization and containers are the technologies that drive the majority of the day to day applications and industry, this course is designed to build upon the knowledge acquired at the undergraduate/graduate level on Operating Systems and familiarise students with virtualization and container technologies.

Prerequisite: Operating Systems

Course Outcomes: After the completion of the course the student will be able to

CO 1	Understand the basics of virtualization technology, architecture, limitations and applications.
CO 2	Apply Networking Principles to setup virtual machines and connect to the network
CO 3	Understand the basics of VM life cycle, VM migrations, VM scheduling and load balancing
CO 4	Understand Container fundamentals including how to configure and set up a container
CO 5	Understand the basics of security, troubleshooting and monitoring aspects in container technology
CO 6	Apply the knowledge in Virtualization and docker to setup VM and dockers.

Mapping of course outcomes with program outcomes

	PO 1	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO 10**	**PO 11**	**PO 12**
CO 1	2	1										
CO 2	3	2	1	1			1					
CO 3	2	1					1					
CO 4	2	1					1					
CO 5	2	1					1					
CO 6	3	2	1	1			1					

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination
	1	2	
Remember	10	10	10
Understand	20	20	20
Apply	20	20	30
Analyse			
Evaluate			
Create			

Mark distribution

Total Marks	CIE	ESE	ESE Duration
100	40	60	3 hours

Continuous Internal Evaluation Pattern:

Attendance	: 8 marks
Continuous Assessment Test (2 numbers)	: 20 marks
Assignment/Quiz/Course project	: 12 marks

End Semester Examination Pattern: There will be two parts; Part A and Part B. Part A contains 10 compulsory short answer questions, 2 from each module. Each question carries 3 marks. Part B contains 2 questions from each module of which student should answer any one. Each question can have maximum 2 sub-divisions and carry 6 marks.

Course Level Assessment Questions

Course Outcome 1 (CO1):

(a) Explain the need and applications of virtualization.

(b) Describe the hypervisor architecture.

(c) Mention tools and technologies used in virtualization.

Course Outcome 2 (CO2)

(a) Describe IP addressing

(b) Explain the concept of paging and virtual memory

Course Outcome 3(CO3):

(a) Describe the VM life cycle.

(b) Explain VM provisioning, VM scheduling and load balancing

(c) Write in detail the KVM architecture and commands

Course Outcome 4 (CO4):

(a) Discuss the container fundamentals and different container technologies.

(b) Explain the container orchestration and clustering.

Course Outcome 5 (CO5):

(a) Discuss the concepts of security and isolution in containers.

(b) Explain troubleshooting, monitoring and alerting in containers.

Course Outcome 6 (CO6):

(a) Explain how to configure and set up virtual machines.

(b) Describe the configuring and setting up of containers.

Model Question paper

Reg No.:________________		Name:____________________________________ _
APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY FIRST SEMESTER M.C.A.DEGREE EXAMINATION, MODEL QUESTION PAPER		
20MCA168 VIRTULAISATION AND CONTAINERS		
Max. Marks: 60		Duration: 3 Hours

PART A			
		Answer all questions, each carries 3 marks.	Marks
1		Explain the different types of hypervisors.	(3)
2		What are the types of virtualization ?	(3)
3		Explain the difference between private IP and Public IP.	(3)
4		What is virtual memory.	(3)
5		Explain any three commands in KVM and their uses.	(3)
6		What do you mean by virtual machine migration?	(3)
7		Write a short note on the container creation process.	(3)
8		Explain virtual networking in containers.	(3)
9		How are the running containers controlled	(3)
10		Explain troubleshooting in container technologies.	(3)
PART B			
Answer any one question from each module. Each question carries 6 marks.			
Module I			
11		Briefly explain the hypervisor architecture.	(6)
OR			
12		Explain difference between virtualization and virtual computing.	(6)
Module II			
13		Briefly explain the different modes in networking can be set up in virtual machines.	(6)
OR			
14		Explain how paging is important in the context of virtualization.	(6)

Module III			
15		Briefly explain the VM life cycle.	(6)
OR			
16		Describe any VM scheduling algorithm.	(6)
Module IV			
17		Explain the Breadth First Search algorithm with a suitable example.	(6)
OR			
18		Compare virtual machines and containers	(6)
Module V			
19		Explain the importance of security and isolation in Docker.	(6)
OR			
20		Explain the business context of containers	(6)

Syllabus

Module 1 *(10 Hours)*: Physical and virtual machines, Traditional and virtual computing, Understanding virtualization, Need and Applications of virtualization, Limitations, Simulations and Emulations, Challenges in Virtualized environment, tools and technologies in virtualized environments. Types of Hypervisors, Hypervisor architecture

Module 2 *(8 Hours)*: IP addressing - Private address, Public address, virtual LAN, Memory addressing, Paging, Memory mapping, virtual memory, complexities and solutions of memory virtualization

Module 3 *(14 Hours)***:** VM lifecycle, Process and system level VMs, VM configurations, VM migrations, Migration types and process, VM provisioning, Scaling,VM scheduling, Load balancing: Significance, Types and Algorithms. Case study : KVM, KVM architecture, KVM commands

Module 4 *(10 Hours)***:** Container fundamentals, Containers versus virtual machines, Different container technologies, Configuring a container engine, Container virtual networking, Container orchestration and clustering, Images and containers. Case study : Docker

Module 5 *(6 Hours)***:** Working with remote repositories, Security and isolation, Troubleshooting, Monitoring and alerting, Controlling running containers, Containers in a business context

References

1. Chris Wolf , Erick M. Halter, ***Virtualization: From the Desktop to the Enterprise***, APress 2005.

2. Kumar Reddy, Victor Moreno, ***Network virtualization***, Cisco Press, July, 2006.

3. James E. Smith, Ravi Nair, ***Virtual Machines: Versatile Platforms for Systems and Processes***, Elsevier/Morgan Kaufmann, 2005

4 Matthew Portnoy, ***Virtualization Essentials***, Wiley; Second edition (2016)

5. Sean P. Kane, Karl Matthias, ***Docker: Up & Running - Shipping Reliable Containers in Production***, Second Edition, O'Reilly

Web References

1. https://www.linux-kvm.org/page/Main_Page
2. https://docs.docker.com/get-started/

Course Contents and Lecture Schedule

No	Topic	No. of Lecture Hours
1	**Introduction**	
1.1	Physical and virtual machines	1
1.2	Traditional and virtual computing	1
1.3	Understanding virtualization	1
1.4	Need, Applications and Limitations of virtualization	2
1.5	Simulations and Emulations	1
1.6	Challenges in Virtualized environment	1
1.7	Tools and technologies in virtualized environments	1
1.8	Types of Hypervisors	1
1.9	Hypervisor architecture	1
2	**Network Virtualization**	
2.1	IP addressing	1
2.2	Private address and Public Addresses	1
2.3	Virtual LAN	1
2.4	Memory addressing	1
2.5	Paging	1
2.6	Memory mapping	1
2.7	Virtual memory	1
2.8	Complexities and solutions of memory virtualization	1

No	Topic	No. of Lecture Hours
3	**Virtual Machine**	
3.1	VM lifecycle	1
3.2	Process and system level VMs	1
3.3	VM configurations	1
3.4	VM migrations	1
3.5	Migration types and process	1
3.6	VM provisioning	1
3.7	Scaling,VM scheduling	2
3.8	Load balancing: Significance	1
3.9	Types and Algorithms	1
3.10	Case study : KVM	1
3.11	KVM architecture	1
3.12	KVM commands	2
4	**Containers**	
4.1	Container fundamentals	1
4.2	Containers versus virtual machines	1
4.3	Different container technologies	1
4.4	Configuring a container engine	1
4.5	Container virtual networking	1
4.6	Container orchestration and clustering	1
4.7	Images and containers	1
4.8	Case study : Docker	3
5	**Security and Management**	
5.1	Working with remote repositories	1
5.2	Security and isolation	1
5.3	Troubleshooting	1
5.4	Monitoring and alerting	1
5.5	Controlling running containers	1
5.6	Containers in a business context	1

20MCA172	ADVANCED OPERATING SYSTEMS	CATEGORY	L	T	P	CREDIT
		ELECTIVE	3	1	0	4

Preamble: This course intends to provide insight into more Advanced Operating Systems. Detailed discussion on various concepts like process synchronization, mutual exclusion, resource sharing, concurrency control and security are discussed at algorithm level. Various kinds of advanced operating systems like Distributed Systems, Multiprocessor systems, and Database Systems are included to the level possible within the scope of a single course. More detailed treatment can be done through seminars , assignments and talks by eminent external experts .

Prerequisite: Basic concepts of desktop computer operating systems.

Course Outcomes: After the completion of the course the student will be able to

CO 1	Identify synchronization problems in operating systems and issues in distributed systems.
CO 2	Explain classification of mutual exclusion algorithms and security violations.
CO 3	Explain the design of distributed shared memory and issues in load distribution.
CO 4	Explain design issues and synchronization in multiprocessor systems.
CO 5	Explain synchronization and concurrency control in database systems.

Mapping of course outcomes with program outcomes

	PO 1	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO 10**	**PO 11**	**PO 12**
CO 1	2	2			2				1			
CO 2	2	1							1			
CO 3	2	1							1			
CO 4	2	1							1			
CO 5	2	2			1		1		1			

Assessment Pattern

Bloom's Category	**Continuous Assessment Tests**		**End Semester Examination**
	1	**2**	
Remember	10		10
Understand	20	20	20
Apply	20	20	20
Analyse		10	10
Evaluate			
Create			

Mark distribution

Total Marks	CIE	ESE	ESE Duration
100	40	60	3 hours

Continuous Internal Evaluation Pattern:

Attendance	: 8 marks
Continuous Assessment Test (2 numbers)	: 20 marks
Assignment/Quiz/Course project	: 12 marks

End Semester Examination Pattern: There will be two parts; Part A and Part B. Part A contains 10 compulsory short answer questions, 2 from each module. Each question carries 3 marks. Part B contains 2 questions from each module of which student should answer any one. Each question can have maximum 2 sub-divisions and carry 6 marks

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Explain synchronization using semaphore.
2. Classify Advanced operating systems.
3. Illustrate limitation of Lamports clocks.

Course Outcome 2 (CO2)

1. Explain some of the algorithms for mutual exclusion.
2. Explain potential security violations.
3. Compare the Lamport's algorithm and Rickart-Agarwala algorithm.

Course Outcome 3(CO3):

1. Explain major design issues and building mechanisms in Distributed file systems.
2. Explain important algorithms for implementing DSM.
3. Explain issues in load distribution.

Course Outcome 4 (CO4):

1. Explain system architecture of Multiprocessor systems.
2. Explain design issues in Database Multiprocessor Systems.
3. Explain how virtualization is implemented.

Course Outcome 5 (CO5):

1. Explain Lock based algorithms for concurrency control in Database Systems.
2. Illustrate Timestamp based algorithms for concurrency control in Database Systems.
3. Explain design issues in Database Systems.

Model Question paper

Part A

1. Categorize various advanced operating systems.
2. Illustrate synchronization using semaphore.
3. Explain potential security violations.
4. Explain requirements of mutual exclusion.
5. What is the difference between load balancing and load sharing?
6. Which are the major components of a load distributing algorithm?
7. Explain the interconnection network in multiprocessors?
8. Explain the structure of Multiprocessor of Operating Systems.
9. Explain what is meant by serializability.
10. What is meant by Log equivalence? (3 x 10 = 30 Marks)

Part B

Module 1

11. Identify any six issues that are common with Distributed systems. [6 Marks]

OR

12. Write a note on the following
    a. mutex [3 marks]
    b. semaphore [3 marks]

Module 2

13. Write short notes on
    a. Rickart-Agarwala Algorithm [3 Marks]
    b. Lamport’s algorithm. [3 Marks]

OR

14. Explain any six Design Principles for Secure Systems. [6 Marks]

Module 3

15. Identify major design issues in Distributed File systems.

[6 Marks]

OR

16. Write any two algorithms for implementing DSM

[6 Marks]

Module 4

17. Explain Multiprocessor System Architectures and Interconnection Networks.

[6 Marks]

OR

18. Discuss the synchronization of processes in Multiprocessors.

[6 Marks]

Module 5

19. Explain the basic Synchronization Primitives for Concurrency Control in Database systems.

[6 Marks]

OR

20. Write and explain an optimistic algorithm for concurrency control in database systems.

[6 Marks]

Syllabus

Module	Contents	Hours
I	**Overview**: Functions of Operating System –Design Approaches –Types of Advanced Operating Systems. **Synchronization Mechanisms**: Concept of Processes and Threads –The Critical Section Problem – Other Synchronization Problems:– Monitor –Serializer – Path Expressions. **Distributed Operating Systems**:- Issues in Distributed Operating System – Communication Networks And Primitives –Lamport's Logical clocks – Causal Ordering of Messages.	10
II	**Distributed Mutual Exclusion**:- Classification - Requirements – Measuring Performance – Lamport's Algorithm – Rickart-Agarwala Algorithm – Suzuki-Kasami's Broadcast Algorithm. : **Security** Potential Security Violations – Design Principles for Secure Systems –The Access Matrix Model and Implementation- The Access Control list Method.	10
III	**Distributed Resource Management**: Mechanisms for building Distributed File Systems – Design Issues - Distributed Shared Memory – Algorithms for Implementing Distributed Shared memory – Issues in Load Distributing – Components of Load Distributing Algorithm – Sender-Initiated Algorithm – Receiver- Initiated Algorithm.	10
IV	**Multiprocessor Operating Systems**: Basic Multiprocessor System Architectures – Interconnection Networks – Structures – Design Issues – Threads – Process - Synchronization – Processor Scheduling – Memory Management – Virtualization – Types of Hypervisors – Paravirtualization – Memory Virtualization – I/O Virtualization.	8
V	**Database Systems**: Problem of Concurrency Control – Serializability – Basic Synchronization Primitives for Concurrency Control – Lock-Based Algorithms – Time-Stamp Based Algorithms – Optimistic Algorithms.	10

Textbooks:

1. Mukesh Singhal and Niranjan G. Shivaratri, "***Advanced Concepts in Operating Systems***
   – Distributed, Database, and Multiprocessor Operating Systems", Tata McGraw-Hill, 2001.
2. Andrew S. Tanenbaum, "***Modern Operating Systems***", 3rd Edition, Prentice Hall, 2012.

Reference Books:

1. Pradeep K Sinha, "***Distributed Operating Systems: Concepts and Design***", Prentice Hall of India, 2007.
2. George Coulouris, Jean Dollimore,Tim Kindberg, Gordon Blair, ***"Distributed Systems,Concepts and Design"***, 5th Edtn,Pearson, 2019
3. https://www.classcentral.com/course/udacity-advanced-operating-systems-1016
4. https://www.my-mooc.com/en/mooc/advanced-operating-systems--ud189/

Course Contents and Lecture Schedule

No	**Topic**	**No. of Lectures**
1	**Introduction**	
1.1	**Overview**: Functions of Operating System –Design Approaches –Types of Advanced Operating Systems.	2
1.2	**Synchronization Mechanisms**: Concept of Processes and Threads – The Critical Section Problem – Other Synchronization Problems:– Monitor –Serializer – Path Expressions.	4
1.3	**Distributed Operating Systems**:- Issues in Distributed Operating System – Communication Networks And Primitives –Lamport's Logical clocks – Causal Ordering of Messages	4
2		
2.1	**Distributed Mutual Exclusion**:- Classification - Requirements – Measuring Performance – Lamport's Algorithm –	2
2.2	. Rickart-Agarwala Algorithm – Suzuki- Kasami's Broadcast Algorithm.	3
2.3	**Security :**Potential Security Violations – Design Principles for Secure Systems –The Access Matrix Model and Implementation- The Access Control list Method	5

No	Topic	No. of Lectures
3		
3.1	**Distributed Resource Management**: Mechanisms for building Distributed File Systems – Design Issues.	3
3.2	**Distributed Shared Memory** – Algorithms for Implementing Distributed Shared memory	3
3.3	**Load Distribution** : Issues in Load Distributing – Components of Load Distributing Algorithm – Sender- Initiated Algorithm – Receiver-Initiated Algorithm.	4
4		
4.1	**Multiprocessor Operating Systems**: Basic Multiprocessor System Architectures – Interconnection Networks – Structures – —.	3
4.2	Design Issues – Threads – Process Synchronization - Processor Scheduling – Memory Management	3
4.3	Virtualization – Types of Hypervisors – Paravirtualization – Memory Virtualization – I/O Virtualization.	2
5		
5.1	**Database Systems**: Problem of Concurrency Control – Serializability – Basic Synchronization Primitives for Concurrency Control – Lock-Based Algorithms-	5
5.2	Time-Stamp Based Algorithms	3
5.3	Optimistic Algorithms	2

20MCA182	BUSINESS MANAGEMENT	CATEGORY		L	T	P	CREDIT
		ELECTIVE		3	1	0	4

Preamble: The primary aim of this course is to understand basic principles of management and accounting. In our day to day life managers will have to manage so many resources in the present day complex business environment. By effective and efficient management the goals of the organisation can be attained. This course is intended to give an idea regarding managing the resources for the effective performance of the organisation and decision making in everyday life. Basic idea regarding book keeping and accounting is also required for managers for taking decisions.

Prerequisite: NIL

Course Outcomes: After the completion of the course the student will be able to

CO 1	Understand management as a process.
CO 2	Critically analyse and evaluate management theories and practices
CO 3	Perform planning and organising for an organisation
CO 4	Do staffing and related human resource development function
CO 5	Take proper decisions to get competitive advantage
CO 6	Understand basic concepts in book keeping and accounting.

Course Outcomes: After the completion of the course the student will be able to

	PO 1	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO 10**	**PO 11**	**PO 12**
CO 1	3							2				
CO 2		3										
CO 3			3									3
CO 4											3	
CO 5					3	2						
CO 6	3							3				

Assessment Pattern

Bloom's Category	**Continuous Assessment Tests**		**End Semester Examination**
	1	**2**	
Remember	10	10	20
Understand	20	20	20
Apply	20	20	20
Analyse			
Evaluate			

Create			

Mark distribution

Total Marks	CIE	ESE	ESE Duration
100	40	60	3 hours

Continuous Internal Evaluation Pattern:

Attendance	: 8 marks
Continuous Assessment Test (2 numbers)	: 20 marks
Assignment/Quiz/Course project	: 12 marks

End Semester Examination Pattern: There will be two parts; Part A and Part B. Part A contain 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which student should answer any one. Each question can have maximum 2 sub-divisions and carry 6 marks.

Course Level Assessment Sample Questions

Course Outcome CO1:

Describe various functions of management.

Course Outcome CO2 :

Explain different theories of management thought.

Course Outcome CO3:

Illustrate different steps in planning.

Course Outcome CO4:

Describe different types of training methods for employees in an organisation.

Course Outcome CO5:

Explain the decision process in an organisation with case example.

Course Outcome CO6:

Explain the procedure of preparation of balance sheet with a simple example.

Model Question Paper

			Total Pages:
Reg No.:________________		Name:____	
APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY SECOND SEMESTER M.C.A. DEGREE EXAMINATION			
Course Code: 20MCA182			
Course Name: BUSINESS MANAGEMENT			
Max. Marks: 60			Duration: 3 Hours
PART A			
		Answer all questions, each carries3 marks.	Marks
1		Define management. What are the levels of management?	(3)
2		Distinguish between efficiency and effectiveness in management.	(3)
3		Explain system approach in management.	(3)
4		Illustrate different types of plans	(3)
5		Explain matrix form of organisation.	(3)
6		What is meant by job analysis?	(3)
7		Explain bench marking.	(3)
8		What is product life cycle?	(3)
9		Explain the rules of debit and credit.	(3)
10		Explain the advantages of accounting softwares.	(3)
PART B			
Answer six questions, one full question from each module and carries 6 marks.			
Module I			
11		What are the different roles that managers play in an organisation?	(6)
OR			
12		Explain the major contributions of F W Taylor to scientific management.	(6)
Module II			
13		Explain various steps involved in planning with a case example.	(6)
OR			
14		Explain any 3 types of organisation structures.	(6)

Module III			
15		Explain various steps involved in selection of employees for an organisation.	(6)
OR			
16		Describe different types of training methods for employees in an organisation.	(6)
Module IV			
17		Illustrate the decision process in an industry by giving different steps involved in it.	(6)
OR			
18		Explain the marketing mix elements with a case example.	(6)
Module V			
19		What is a Journal? Explain the rules of journalising	(6)
OR			
20		What are final accounts? Explain the procedure of preparing balance sheet with a simple example.	(6)

Syllabus

Module I

Introduction to Management: Basic Managerial Concepts, Levels of management, Managerial Skills, Managerial role. Management functions- Planning, Organising, Staffing, leading and Controlling.

Early Contributions in Management: Management thought - Classical approach, scientific management, contributions of Taylor, Gilbreths, Fayol's 14 principles of management.

Human relation approach - contribution of Elton Mayo Systems approach - organization as an open system and Contingency approach

Module II

Planning: Nature and importance of planning, types of plans - Steps in planning, Levels of planning - The Planning Process - MBO definition and process, SWOT Analysis, importance.

Organising : Nature of organizing,-span of control in management, factors affecting span of control- Authority and responsibility.

Organisation structure - Formal and informal, Types of organization structure line, line and staff, functional, divisional, project, matrix, virtual form of organisations.

Module III

Staffing and related HRD Functions: meaning, nature, staffing process, Job analysis and manpower planning, job description and job specification, Recruitment & selection, selection process, Tests and interviews. Training and development - concept and methods ,Performance appraisal- concept and methods.

Module IV

Managerial Decision Making and controlling : Decision making –types of decisioins, decision making process, Decision Making Tools, Importance of controlling, Techniques of controlling- Break Even Analysis, Budgetary Control - Benchmarking –importance and limitations of benchmarking, Six Sigma importance, limitations and process of six sigma, Total Quality Management- Introduction to marketing management-Marketing mix- product life cycle

Module V

Book- Keeping and Accountancy -Elements of Double Entry -Book- Keeping - rules for journalizing -Ledger accounts –Cash book- – Trial Balance- Method of Balancing accounts- the journal proper (simple problems). Final accounts: Preparation of trading and profit and loss Account- Balance sheet (with simple problems) - Introduction to Accounting packages (Description only)

References

1. L M Prasad, "***Principles of Management***", Sultan Chand & Sons, 8th Edition (2010)
2. Peter F Drucker, "***The Practice of Management***", Butterworth-Heinemann publication, 2nd Edition (2007)
3. Harold Koontz and Heinz Weihrich, "***Essentials of Management***", McGraw Hill Education, 10th Edition (2015).
4. Robbins and Coulter, ***Management***, Pearson Education 13th Edition, 2016,
5. R N Gupta, "***Principles of Management***", S. Chand & Company Ltd., (2010)
6. Tripathi, "***Principles of Management***", McGraw Hill Education, 5th Edition (2012)
7. ***Double Entry book Keeping*** – Batliboi
8. ***A Systematic approach to Accounting***: Dr K.G. Chandrasekharan Nair

Suggested MOOCs

1. Management Functions http://nptel.ac.in/courses/122108038/
2. Leadership http://nptel.ac.in/courses/110105033/33

Course Contents and Lecture Schedule

No	Topic	No. of Lectures
1	Introduction to Management: Basic Managerial concepts	2
1.1	Levels of management, Managerial Skills	2
1.2	Management roles	1
1.3	Management functions	2
1.4	Early Contributions in Management: Management thought - Classical approach, scientific management, contributions of Taylor, Gilbreths, Fayol's 14 principles of management. Human relation approach - contribution of Elton Mayo Systems	3
2	Planning: Nature and importance of planning, types of plans - Steps in planning, Levels of planning - The Planning Process	2
2.1	MBO definition and process, SWOT Analysis, importance.	2
2.2	Organising : Nature of organizing,-span of control in management, factors affecting span of control- authority and responsibility.	2
2.3	Organisation structure - Formal and informal, Types of organization structure line, line and staff, functional, divisional, project, matrix, virtual form of organisations	2
3	Staffing and related HRD Functions: meaning, nature, staffing process.	2
3.1	Job analysis and manpower planning, job description and job specification	2
3.2	Recruitment & selection, selection process, Tests and interviews. Training and development - concept and methods	3
3.3	Performance appraisal - concept and methods.	2
4	Managerial Decision Making and controlling : Decision making –types of decisions, decision making process, Decision Making Tools.	2
4.1	Importance of controlling, Techniques of controlling- Break Even Analysis, Budgetary Control	2
4.2	Benchmarking –importance and limitations of benchmarking	2
4.3	Six Sigma importance, limitations and process of six sigma,	2
4.4	Total Quality Management-	2
4.5	Introduction to marketing management-Marketing mix- product life cycle	2

No	Topic	No. of Lectures
5	Book- Keeping and Accountancy -Elements of Double Entry -Book-Keeping	2
5.1	Rules for journalizing -Ledger accounts –Cash book-	3
5.2	Trial Balance- Method of Balancing accounts- (simple problems). Final accounts: Preparation of trading and profit and loss Account- Balance sheet (with simple problems)	3
5.3	Introduction to Accounting packages.	2

20MCA184	EMBEDDED SYSTEMS	CATEGORY	L	T	P	CREDIT
		ELECTIVE	3	1	0	4

Preamble: This course introduces students to the basic concepts behind Embedded Systems. It helps the students to understand the various techniques involved in embedded system design and development.

Prerequisite:

- 20MCA103 Digital Fundamentals & Computer Architecture.
- 20MCA107 Advanced Software Engineering.
- Basic knowledge of the subjects Operating Systems and System Software.

Course Outcomes: After the completion of the course the student will be able to

CO 1	Understand the basic concepts of Embedded Systems and its Applications.
CO 2	Demonstrate the role of individual components involved in a typical embedded system.
CO 3	Learn about the co-design approach for embedded hardware and firmware development.
CO 4	Understand the concepts involved in Embedded System Design and development Process.
CO 5	Learn about techniques used in the Integration and Testing of Embedded Hardware and Firmware.
CO 6	Understand the basic concepts of RTOS based Embedded System Design.

Mapping of course outcomes with program outcomes

	PO 1	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO 10**	**PO 11**	**PO 12**
CO 1	2											
CO 2	3	3										
CO 3	3	3			3							
CO 4			3	2								
CO 5		3										
CO 6		3										

Assessment Pattern

Bloom's Category	**Continuous Assessment Tests**		**End Semester Examination**
	1	**2**	
Remember(K1)	10	10	10
Understand(K2)	20	20	20
Apply(K3)	20	20	30
Analyse(K4)			
Evaluate(K5)			
Create(K6)			

Mark distribution

Total Marks	**CIE**	**ESE**	**ESE Duration**
100	40	60	3 hours

Continuous Internal Evaluation Pattern:

Attendance	: 8 marks
Continuous Assessment Test (2 numbers)	: 20 marks
Assignment/Quiz/Course project	: 12 marks

End Semester Examination Pattern: There will be two parts; Part A and Part B. Part A contains 10 compulsory short answer questions, 2 from each module. Each question caries 3 marks. Part B contains 2 questions from each module of which student should answer any one. Each question can have maximum 2 sub-divisions and carry 6 marks.

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Define Embedded System.
2. Illustrate the major Applications of Embedded System.
3. List out the classifications of Embedded System.

Course Outcome 2 (CO2):

1. Illustrate the components of an embedded System with the help of relevant diagram.
2. Explain about the processor Embedded into a System.

Course Outcome 3 (CO3):

1. Describe the Fundamental Issues in Hardware Software Co-Design.
2. Explain UML with the help of an example.

Course Outcome 4 (CO4):

1. Describe any three Digital Electronic Components used in the embedded Hardware development.
2. Explain about Embedded Firmware Design Approaches.

Course Outcome 5 (CO5):

1. Explain any one technique used for the Integration of Hardware and Firmware.
2. List out the techniques used for the Testing of Embedded Systems.

Course Outcome 6(CO6):

1. Define RTOS.
2. Describe How you will Choose an RTOS.

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

Model Question Paper

Course Code: 20MCA184

Course Name: Embedded Systems

Max. Marks: 60		Duration: 3 Hours

PART A

		Answer all questions, each carries 3 marks.	Marks
1		Define embedded computing system? Write two functionalities of an embedded system.	(3)
2		What are the building blocks and devices of hardware in an embedded system?	(3)
3		Describe the Design Process in Embedded System.	(3)

4		Explain about Formalism of System Design.	(3)
5		Illustrate Data Flow Graph Model with the help of relevant diagram.	(3)
6		Define State Machine Model with the help of suitable example.	(3)
7		Describe the Analog Electronic Components in Embedded Hardware Design.	(3)
8		Illustrate the Super Loop Based firmware development approach.	(3)
9		How will you Test Embedded Systems?	(3)
10		Define Real Time Operating System.	(3)
PART B			
Answer any one question from each module. Each question carries 6 marks.			
Module I			
11		Differentiate Embedded system and General Computing System.	(6)
OR			
12		Explain about the Classification of Embedded systems.	(6)
Module II			
13		Explain in detail about Design Challenges in Embedded System Design.	(6)
OR			
14		Explain about the Hardware-Software Co-Design in an Embedded System.	(6)
Module III			
15		With the help of suitable diagrams explain about UML Building Blocks.	(6)
OR			
16		Describe the Fundamental Issues in Hardware Software Co-Design.	(6)
Module IV			
17		Explain about any three Digital Electronic Components in Embedded systems with the help of suitable diagrams.	(6)
OR			
18		Explain about Embedded Firmware Development Languages.	(6)
Module V			

19		Explain in detail about the commonly used techniques for the Integration of Hardware and Firmware.	(6)
		OR	
20		A lot of factors needs to be analysed carefully before making a decision of choosing an RTOS. Justify.	(6)
		****	

Syllabus

Module 1

Introduction to Embedded Systems: Embedded system, Embedded system Vs General Computing System, Processor Embedded into a System, Embedded Hardware units and devices in a system, Embedded Software in a System, Introduction to embedded system design, Classification of Embedded systems, Skills Required for an embedded system Designer, Examples of the Embedded Systems. Major Application Areas of Embedded Systems, Purpose of Embedded Systems.

Module 2

Embedded System Design and development Process: Embedded System-On-Chip (SoC) and Use of VLSI Circuit Design Technology, Build Process for embedded systems, Design Process in Embedded System, Design Challenges in Embedded System Design, Hardware-Software Co-Design in an Embedded System, Formalism of System Design.

Module 3

Hardware Software Co-Design and Program Modelling: – Fundamental Issues in Hardware Software Co-Design, Computational Models in Embedded Design - Data Flow Graph Model, Control Data Flow Graph, State Machine Model, Sequential Program Model, Concurrent / Communicating Process Model, Object oriented model, UML.

Module 4

Design and Development of Embedded Product:

Embedded Hardware Design and Development: - Analog Electronic Components, Digital Electronic Components, VLSI and Integrated Circuit Design.

Embedded Firmware Design and Development: - Embedded Firmware Design Approaches, Embedded Firmware Development Languages.

Module 5

Integration and Testing of Embedded Hardware and Firmware: - Integration of Hardware and Firmware, Testing Embedded Systems.

RTOS based Embedded System Design: - Basic operating system services, Introduction to Real Time Operating System(RTOS), RTOS Task-Scheduling models, How to Choose an RTOS.

Text Books

1. Raj Kamal, Embedded Systems: Architecture, Programming and Design, Third Edition, McGraw Hill Education (India), 2014.
2. Shibu K.V., Introduction to Embedded Systems, McGraw Hill Education (India), 2009.

Reference Books

1. J Staunstrup and Wayne Wolf, Hardware / Software Co-Design: Principles and Practice, Prentice Hall.
2. Jean J. Labrose, Micro C/OS II: The Real Time Kernel, 2e, CRC Press, 2002.
3. Steave Heath, Embedded System Design, Second Edition, Elsevier.
4. Wayne Wolf , Computers as Components-Principles of Embedded Computer System Design, Morgan Kaufmann publishers, Third edition, 2012.

Web Resources

1. https://nptel.ac.in/courses/108/102/108102045/
2. https://www.coursera.org/learn/embedded-software-hardware.
3. https://www.edx.org/course/embedded-systems-shape-the-world-multi-threaded-in.

Course Contents and Lecture Schedule

Topic	**No. of lectures**
Module 1	**9 hrs.**
Introduction to Embedded Systems: Embedded system, Embedded system Vs General Computing System, Processor Embedded into a System.	2
Embedded Hardware units and devices in a system, Embedded Software in a System, Introduction to embedded system design, classification of Embedded	

systems, Skills Required for an embedded system Designer, Examples of the Embedded Systems.	5
Major Application Areas of Embedded Systems, Purpose of Embedded Systems.	2
Module 2	**10 hrs**
Embedded System Design and development Process: Embedded System-On-Chip (SoC) and Use of VLSI Circuit Design Technology, Build Process for embedded systems.	4
Design Process in Embedded System, Design Challenges in Embedded System Design.	3
Hardware-Software Co-Design in an Embedded System, Formalism of System Design.	3
Module 3	**9 hrs**
Hardware Software Co-Design and Program Modelling: – Fundamental Issues in Hardware Software Co-Design.	2
Computational Models in Embedded Design - Data Flow Graph Model, Control Data Flow Graph, State Machine Model, Sequential Program Model, Concurrent / communicating Process Model, Object oriented model, UML.	7
Module 4	**10 hrs**
Design and Development of Embedded Product: **Embedded Hardware Design and Development:** - Analog Electronic Components, Digital Electronic Components, VLSI and Integrated Circuit Design.	5
Embedded Firmware Design and Development: - Embedded Firmware Design Approaches, Embedded Firmware Development Languages.	5
Module 5	**10 hrs**
Integration and Testing of Embedded Hardware and Firmware: - Integration of Hardware and Firmware, Testing Embedded Systems.	6
RTOS based Embedded System Design: - Basic operating system services, Introduction to Real Time Operating System(RTOS), RTOS Task-Scheduling models, How to Choose an RTOS.	4

20MCA186	COMPUTER GRAPHICS	CATEGORY	L	T	P	CREDIT
		ELECTIVE	3	1	0	4

Preamble: This subject intends to provide an overview of the foundations of Computer Graphics rendering. Special emphasis is laid on modern concepts like Ray Tracing that have already become industry standards for graphics rendering with modern GPUs. Other fundamentals such as colorimetry and radiometry are also introduced in the subject. Although the course is expected to be treated theoretically for evaluation purposes, practical sessions and talks by external experts from the graphics processing industry may be desirable.

Prerequisite:

Fundamentals of computer hardware, Linear Algebra

Course Outcomes: After the completion of the course the student will be able to

CO 1	Apply foundational knowledge in computer graphics to work with Graphics APIs
CO 2	Explain various shape drawing algorithms and transformations.
CO 3	Explain viewing concepts and follow the workflow in computer graphics pipeline.
CO 4	Explain different shading, texture mapping and data structures used in computer graphics.
CO 5	Apply concepts in Raytracing to better understand and design computer graphics models.
CO 6	Apply concepts in colorimetry and radiometry to work with images.

Mapping of course outcomes with program outcomes

	PO 1	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO 10**	**PO 11**	**PO 12**
CO 1	3											
CO 2	3	2	2		3							
CO 3	3											
CO 4	3											
CO 5	3				1		1					
CO 6	3											

Assessment Pattern

Bloom's Category	**Continuous Assessment Tests**		**End Semester Examination**
	1	**2**	
Remember	10	10	10
Understand	20	20	20
Apply	20	20	30
Analyse			
Evaluate			
Create			

Mark distribution

Total Marks	**CIE**	**ESE**	**ESE Duration**
100	40	60	3 hours

Continuous Internal Evaluation Pattern:

Attendance	: 8 marks
Continuous Assessment Test (2 numbers)	: 20 marks
Assignment/Quiz/Course project	: 12 marks

End Semester Examination Pattern: There will be two parts; Part A and Part B. Part A contains 10 compulsory short answer questions, 2 from each module. Each question caries 3 marks. Part B contains 2 questions from each module of which student should answer any one. Each question can have maximum 2 sub-divisions and carry 6 marks

Course Level Assessment Questions

Course Outcome 1 (CO1):

4. Explain the workflow in the computer graphics pipeline.
5. How does alpha composition affect image appearance?
6. Explain Eigen vectors and Eigen values?

Course Outcome 2 (CO2)

4. Explain some of the pitfalls of Bresenham line drawing algorithm.
5. Explain Affine transformation and its purpose.
6. Compare the mid-point and Bresenham circle drawing algorithms.

Course Outcome 3(CO3):

4. Explain how anti-aliasing affects the image quality.
5. Explain projective transformation.
6. Explain Field of View.

Course Outcome 4 (CO4):

4. Explain some of the benefits of using Triangle meshes.
5. How is Phong shading different from Artistic Shading?
6. How is texture mapping for rasterized image performed.

Course Outcome 5 (CO5):

4. Explain what makes Ray Tracing a highly system intensive rendering process.
5. How can transparency be achieved using ray tracing.
6. Explain the techniques used in Ray Tracing for shadows.

Course Outcome 6 (CO6):

1. Explain Tonal Reproduction.
2. Write a short note on particle tracing for Lambertian scenes.
3. How can rough and smooth surfaces be modeled?

Model Question paper

Part A

21. Explain the concept and idea behind pixels.
22. Write three major applications of computer graphics.
23. Explain the term antialiasing?
24. Demonstrate how the DDA Line drawing algorithm works with a simple example of your own.
25. What is the purpose of tiling multidimensional arrays.
26. Explain the meaning and purpose of graphics APIs?
27. Explain the concept of ray tracing?
28. Explain the basic concept behind ray-object intersection and how it is established.
29. Explain what is meant by color space and give examples.
30. What is meant by a High Dynamic Range image? [3x10 =30 Marks]

Part B

Module 1

31. Explain the process of converting an analog image to a digital image. [6 Marks]

OR

32. Write a note on the following
    a. RGB Color Space [3 marks]
    b. Matrix Diagonalisation [3 marks]

Module 2

33. Write short notes on
    c. Projective Transformation [3 Marks]
    d. Perspective Projection [3 Marks]

OR

34. Explain a typical Graphics Pipeline. [6 Marks]

Module 3

35. Compare the various graphics APIs citing their advantages and disadvantages.

OR [6 Marks]

36. Explain the various specialized data structures used in computer graphics

[6 Marks]

Module 4

37. Explain how shading and shadowing are achieved using Ray Tracing.

OR [6 Marks]

38. Explain the geometry for graphics hardware.

[6 Marks]

Module 5

39. Explain the methods for Nighttone mapping.

[6 Marks]

OR

40. Explain how accurate direct lighting can be achieved.

[6 Marks]

Syllabus

Module	Contents	Hours
I	**Introduction to computer graphics**: Major Areas and Major Applications, Preliminary discussion on Graphics Pipeline, Numerical Issues, Efficiency and Coding Graphics Programs. **Raster Images**: Raster Devices, Images, Pixels, RGB Color and Alpha Composition. **Fundamentals of Signal Processing** for Images and Sampling Theory(Theoretical understanding only). **Mathematical Foundations of Computer Graphics**: Review of Trigonometry and Geometry, Theoretical foundations of Linear Algebra – Vectors and Matrices, Eigen Values and Eigen Vectors, Matrix Diagonalization(Theoretical understanding only).	9
II	**Fundamentals of shape drawing**:- Line drawing - DDA and Bresenham Algorithms, Circle drawing: Mid Point and Bresenham. **Transformations(2D, 3D)**:, Translation and Affine Transformations, Inverse of Transformation Matrices, Coordinate transformations. **Viewing**: Viewing Transformations, Projective Transformation, Perspective Projection, Field of View **Graphics Pipeline**: Rasterization, Operations, Antialiasing, Culling primitives for efficiency.	11
III	**Surface shading**: Diffuse Shading, Phong Shading, Artistic Shading. **Texture Mapping**: 2D and 3D Mapping, Texture Mapping for Rasterized Triangles, Bump Textures, Displacement Mapping, Shadow Maps. **Data Structures for Graphics**: Triangle Meshes, Scene Graphs, Spatial Data Structures, BSP Tree for visibility, Tiling Multidimensional Arrays. **Graphics APIs**: Intuitive understanding of role of Graphics APIs such as OpenGL, Direct3D(DirectX), Vulkan etc. – No programming required	10
IV	**Ray Tracing**: Basic Ray Tracing Algorithms, Perspective, Computing Viewing Rays, Ray-Object Intersection, Shading, Shadows, Ideal Specular Reflection, Ray Tracing Program, Transparency and Refraction, Instancing, Constructive Solid Geometry, Distribution Ray Tracing. **Using Graphics Hardware**: Introduction, Geometry for Hardware, Processing Geometry using Pixels.	8

Module	Contents	Hours
V	**Light**: Radiometry, Transport Equation, Photometry; **Colors**: Colorimetry, Color Spaces, Chromatic Adaption, Color Appearance; **Tonal Reproduction**: Classification, Dynamic Range, Image Formation, Frequency based Operators, Gradient Domain Operators, Gradient Domain Operators, Spatial Operators, Division, Sigmoids, Night Tonemapping. **Global Illumination**: Particle tracing for Lambertian scenes, Path Tracing, Accurate Direct Lighting. Reflection Models: Real World Materials, Implementing Reflection Models, Specular Reflection models, Smooth layered Model, Rough Layered Model.	11

Textbooks:

1. Peter Shirley, Steve Marschner: ***"Fundamentals of Computer Graphics"***, 4th Edtn. AK Peters, 2015. – All Modules.
2. Donald Hearn and M. Pauline Baker, ***"Computer Graphics"***, 2nd Edtn. PHI, 1996. – Module 2(Fundamentals of Shape Drawing).

Reference Books:

1. Matt Pharr and Greg Humphreys, ***"Physically Based Rendering: From Theory to Implementation"***, 2nd Edtn, Morgan Kaufmann,2010;
2. Gilbert Strang, ***"Introduction to Linear Algebra"***, 4th Edtn, Wellesley-Cambridge Press, 2009
3. William Stallings, ***"Data and Computer Communications"***, 10th Edtn, Pearson, 2013.
4. Vulkan Documentation, https://www.khronos.org/vulkan/
5. OpenGL Documentation, https://www.khronos.org/opengl/
6. Nvidia Developer, ***"Nvidia Ray Tracing Documentation"***, Nvidia Documentation, https://raytracing-docs.nvidia.com/. – Module 3 and 4, Topics on Ray Tracing.

Course Contents and Lecture Schedule

No	Topic	No. of Lectures
1	**Introduction**	
1.1	**Introduction to computer graphics**: Major Areas and Major Applications, Preliminary discussion on Graphics Pipeline, Numerical Issues, Efficiency and Coding Graphics Programs.	2

No	Topic	No. of Lectures
1.2	**Raster Images**: Raster Devices, Images, Pixels, RGB Color and Alpha Composition.	3
1.3	**Fundamentals of Signal Processing** for Images and Sampling Theory(Theoretical understanding only). **Mathematical Foundations of Computer Graphics**: Review of Trigonometry and Geometry, Theoretical foundations of Linear Algebra – Vectors and Matrices, Eigen Values and Eigen Vectors, Matrix Diagonalization(Theoretical understanding only).	4
2	**Shape Drawing, transformations and Viewing**	
2.1	**Fundamentals of shape drawing**:- Line drawing - DDA and Bresenham Algorithms, Circle drawing: Mid Point and Bresenham.	2
2.2	**Transformations(2D, 3D)**:, Translation and Affine Transformations, Inverse of Transformation Matrices, Coordinate transformations.	2
2.3	**Viewing**: Viewing Transformations, Projective Transformation, Perspective Projection, Field of View	3
2.4	**Graphics Pipeline**: Rasterization, Operations, Antialiasing, Culling primitives for efficiency.	4
3	**Shading**	
3.1	**Surface shading**: Diffuse Shading, Phong Shading, Artistic Shading.	3
3.2	**Texture Mapping**: 2D and 3D Mapping, Texture Mapping for Rasterized Triangles, Bump Textures, Displacement Mapping, Shadow Maps.	3
3.3	**Data Structures for Graphics**: Triangle Meshes, Scene Graphs, Spatial Data Structures, BSP Tree for visibility, Tiling Multidimensional Arrays.	4

No	Topic	No. of Lectures
4	**Ray Tracing, Graphics Hardware and APIs**	
4.1	**Ray Tracing**: Basic Ray Tracing Algorithms, Perspective, Computing Viewing Rays, Ray-Object Intersection, Shading, Shadows, Ideal Specular Reflection, Ray Tracing Program, Transparency and Refraction, Instancing, Constructive Solid Geometry, Distribution Ray Tracing.	3
4.2	**Using Graphics Hardware**: Introduction, Geometry for Hardware, Processing Geometry using Pixels.	3
4.3	**Graphics APIs**: Intuitive understanding of role of Graphics APIs such as OpenGL, Direct3D(DirectX), Vulkan etc. – No programming required	2
5	**Radiometry, colorimetry and tones**	
5.1	**Light**: Radiometry, Transport Equation, Photometry; **Global Illumination**: Particle tracing for Lambertian scenes, Path Tracing, Accurate Direct Lighting Reflection Models: Real World Materials, Implementing Reflection Models, Specular Reflection models, Smooth layered Model, Rough Layered Model.	5
5.2	**Colors**: Colorimetry, Color Spaces, Chromatic Adaption, Color Appearance;	3
5.3	**Tonal Reproduction**: Classification, Dynamic Range, Image Formation, Frequency based Operators, Gradient Domain Operators, Gradient Domain Operators, Spatial Operators, Division, Sigmoids, Night Tonemapping.	3

20MCA188	ARTIFICIAL INTELLIGENCE	CATEGORY	L	T	P	CREDIT
		ELECTIVE	3	1	0	4

Preamble: This course introduces the techniques of Artificial Intelligence and analyzes various methods of solving problems using it. The concept of expert system architecture & fuzzy operations are introduced. This course serves as a prerequisite for many advanced courses in Data Science areas.

Prerequisite: Mathematical Foundations for Computing, Advanced Data structures

Course Outcomes: After the completion of the course the student will be able to

CO 1	Apply the steps needed to provide a formal specification for solving the problem.
CO 2	Apply and analyze the different types of control and heuristic search methods to solve problems
CO 3	Understand various Game theory problems& Knowledge structures
CO 4	Formulate knowledge representation and examine resolution in predicate and propositional logic
CO 5	Apply feasible planning and learning techniques to solve non-trial problems
CO 6	Analyze expert systems & fuzzy operations to solve real life problems.

Mapping of course outcomes with program outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	3	3							2			
CO 2	3	3							2			
CO 3	3	3							2			
CO 4	3	3							2			
CO 5	3	3							2			
CO 6	3	3	3				3		2		2	2

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination
	1	2	
Remember(K1)	10	10	10
Understand(K2)	20	20	20
Apply(K3)	20	20	30
Analyse(K4)			
Evaluate(K5)			
Create(K6)			

Mark distribution

Total Marks	CIE	ESE	ESE Duration
100	40	60	3 hours

Continuous Internal Evaluation Pattern:

Attendance	: 8 marks
Continuous Assessment Test (2 numbers)	: 20 marks
Assignment/Quiz/Course project	: 12 marks

End Semester Examination Pattern: There will be two parts; Part A and Part B. Part A contains 10 compulsory short answer questions, 2 from each module. Each question caries 3 marks. Part B contains 2 questions from each module of which student should answer any one.Each question can have maximum 2 sub-divisions and carry 6 marks

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Describe the areas of Artificial intelligence. (K1)

2. List the problem formulations & production characteristics. (K1 & K2)

3. Solve the various problems such as 8 puzzle, Crypt arithmeticetc(K3)

Course Outcome 2 (CO2):

1. Describe search strategies in solving problems. (K1 & K2)

2. List the disadvantages of hill climbing algorithm (K1& K2)

3. Illustrate A* algorithm for the graph (K3)

Course Outcome 3 (CO3):

1. Demonstrate two player Zero sum game (K3)

2. List and explain the knowledge representation methods in AI. (K1&K2)

3. Explain how alpha-beta algorithm works in pruning of branches with an example.(K3)

Course Outcome 4 (CO4):

1. Translate the following sentence to predicate logic (K3)

a) 'All pompeians were Roman'

b)'All Romans were either loyal to Caesar or hated him'.

2. Explain the algorithm to convert WFF to clause.(K1 & K2)

3.Describe about resolution graph in predicate and propositional logic.(K1 & K2)

Course Outcome 5 (CO5):

1. Differentiate between Goal stack and Hierarchical planning in AI. (K1 & K2)

2. Discuss about neural net learning(K1 & K2)

3. List out the steps in genetic learning. (K1 & K2)

Course Outcome 6 (CO6):

1. Specify the components in expert system. (K1 & K2)

2. Solve various fuzzy operations (K3)

3. List out & explain various tools and languages in AI. (K1 & K2)

Model question paper

Part A

1. List the applications areas in AI
2. Solve the following cryptarithmetic problem

   SEND +
   MORE
   MONEY

3. Explain iterative deepening search
4. List the disadvantages of hill climbing
5. Solve a simple two player Zero sum game
6. Explain about conceptual dependency
7. Explain inference rules in FOPL
8. List components of a planning system
9. Give a short note on role of an expert system
10. List various fuzzy operations

(10X3=30 marks)

Part B

11. Consider a water jug problem .You are given two jugs, a 4 gallon and 3 gallons. Neither has any measuring markers on it. There is a pump that can be used to fill the jugs with water. How can you get exactly 2 gallons of water into 4-gallon jug.State the production rule for waterjug problem (6)

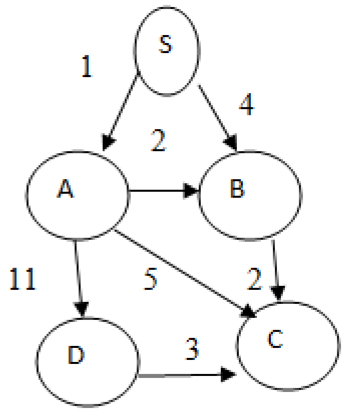
OR

12. Solve missionaries and cannibals problem (6)

13. Explain blind search strategies in detail (6)

OR

14. Explain A* Algorithm for the given graph (6)



Heuristic Value:

S	6
A	2
B	3
C	0
D	5

15. List and explain the knowledge representation methods in AI. (6)

OR

16. Explain how alpha-beta algorithm works in pruning of branches with an example. (6)

17. Explain the algorithm to convert WFF to clause with an example. (6)

OR

18. Explain Neural net and Genetic learning methods in AI (6)

19. Illustrate architecture of an expert system and mention its features. (6)

OR

20. Solve the following using various Fuzzy set operations (6)
A= {0.1/1,0.3/2,0.45/3}
B={0.15/1,0.34/2}

(5X6=30 Marks)

SYLLABUS

Module 1

Introduction to AI and Production Systems:- AI-Problem formulation, Problem Definition - Production systems, Problem characteristics, Production system characteristics , Example AI Problems (8 Puzzle problem, Missionary Cannibals Problem, Crypt arithmetic Problems, block world Problem)

Module 2

Search Strategies : - Blind search strategies -Depth First Search, Breadth First Search, Best First Search, Iterative Deepening Search, Heuristic Search strategies- Admissible Heuristics and examples - Simple Hill Climbing and Steepest Ascending Hill Climbing, Simulated Annealing , A* algorithm.

Module 3

Game playing : Two Player Zero Sum Games, Modelling Two Player Zero Sum Games as search problems, Min-Max Algorithm, Optimizing Min Max Algorithm using α – β cut off, *Knowledge Representation Structures :* Frames, Sematic Networks and Conceptual Dependencies.

Module 4

Knowledge representation using Logic : - First Order Predicate Logic (FOPL), Well Formed Formula(WFF) in FOPL, Inference rules for FOPL, The Clause Form and conversion of WFFs to Clause Form, Resolution- Refutation .*Planning :*- Overview, components of a planning system, Goal stack planning, Hierarchical planning, *Learning :*-Forms of learning, neural net learning & genetic learning

Module 5

Expert systems:–Architecture of expert systems, Roles of expertsystems,Languages and tools – Typical expert system examples.*Fuzzy Logic:* - Fuzzy Variables ,Fuzzy Sets and Fuzzy Set Operations, Typical Examples using FuzzySets.

Text Books

1. Kevin Night and Elaine Rich, "***Artificial Intelligence (SIE)***", McGrawHill-2008.
2. StuartRussel and Peter Norvig "***AI – A Modern Approach***", 2nd Edition, Pearson Education2007.

Reference Books

1.Peter Jackson, "***Introduction to Expert Systems***", 3rd Edition, Pearson Education,2007.
2.Dan W. Patterson, "***Introduction to AI and ES***", Pearson Education,2007.

Course Contents and Lecture Schedule

No	Topic	No. of Lectures
1	***Module I: Introduction to AI***	***9 hrs***
1.1	AI-Problem formulation, Problem Definition -Production systems	
1.2	Production system characteristics	
1.3	AI Problems	
2	***Module II: Search Strategies***	***9 hrs***
2.1	Blind search strategies	
2.2	Heuristics search strategies	
2.3	Simple Hill Climbing and Steepest Ascending Hill Climbing,	
2.4	Simulated anneling	
2.5	A* algorithm	
3	***Module III: Game playing***	***9 hrs***
3.1	Zero sum game	
3.2	Minimax algorithm	
3.3	Alpha beta pruning	
3.4	Knowledge representation structure	
4	***Module IV: Knowledge representation using Logic***	***12 hrs***
4.1	First Order Predicate Logic (FOPL)	
4.2	Well Formed Formula(WFF) in FOPL, Inference rules for FOPL	
4.3	The Clause Form and conversion of WFFs to Clause Form	
4.4	Resolution	
4.5	Planning	
4.6	Learning	
5	***Module V: APPLICATIONS***	***6 hrs***
5.1	Expert system Architecture	
5.2	Fuzzy logic operations	
5.3	Languages and tools	

20MCA192	IPR AND CYBER LAWS	CATEGORY	L	T	P	CREDIT
		ELECTIVE	3	1	0	4

Preamble: This course intends to provide insight into Intellectual Property Rights and Cyber Laws. It includes detailed discussion on various intellectual property rights, procedures to apply for copyrights &patents, legalities of intellectual property to avoid plagiarism and other IPR related crimes. Effectiveness of cyber-laws and other countermeasures against cybercrime and cyber warfare are discussed in detail. Various kinds of Intellectual Property issues in cyberspace and the growth and development of the law in this regard are included to the level possible within the scope of a single course. More detailed treatment can be done through seminars, assignments and talks by eminent external experts including industry.

Prerequisite: General awareness on internet essentials, web technologies, e-commerce.

Course Outcomes: After the completion of the course the student will be able to

CO 1	Explain the fundamentals of IPR and patents.
CO 2	Apply intellectual property related tools such as trademark and copyright to real problems.
CO 3	Discuss Industrial designs, trade secret and geographic Indications.
CO 4	Describe laws governing cyberspace and analyze the role of Internet Governance in framing policies for Internet security.
CO 5	Discuss different types of cybercrimes and penalties under IT Act.

Mapping of course outcomes with program outcomes

	PO 1	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO 10**	**PO 11**	**PO 12**
CO 1	3	2	2	1		1						
CO 2	3	3	2	1		1						
CO 3	3	2	1	1								
CO 4	2	2	1			1						
CO 5	2	2	1	1		1						

Assessment Pattern

Bloom's Category	**Continuous Assessment Tests**		**End Semester Examination**
	1	**2**	
Remember	10	10	10
Understand	20	30	25
Apply	20	10	25
Analyse			
Evaluate			
Create			

Mark distribution

Total Marks	CIE	ESE	ESE Duration
100	40	60	3 hours

Continuous Internal Evaluation Pattern:

Attendance	: 8 marks
Continuous Assessment Test (2 numbers)	: 20 marks
Assignment/Quiz/Course project	: 12 marks

End Semester Examination Pattern: There will be two parts; Part A and Part B. Part A contains 10 compulsory short answer questions, 2 from each module. Each question carries 3 marks. Part B contains 2 questions from each module of which student should answer any one. Each question can have maximum 2 sub-divisions and carry 6 marks

Course Level Assessment Questions

Course Outcome 1 (CO1):

4. Discuss the need for protection of intellectual property.
5. Explain TRIPS Agreement.
6. Illustrate types of patent applications.

Course Outcome 2 (CO2)

4. Explain Trademark Infringement and Protection of trademarks.
5. Explain the rights conferred by copyright, registration and ownerships of copyrights.
6. Discuss about software copyright.

Course Outcome 3(CO3):

4. Discuss the need for protection of design and explain Design Act, 2000.
5. Explain basic concepts of Geographic Indications such as filing, granting and Protection of geographic indications.
6. Describe the procedure of discovering and protecting of trade secret.

Course Outcome 4 (CO4):

4. Explain the need for cyber laws.
5. Discuss protection of copyright on cyberspace.
6. Explain ISP in cyberspace.

Course Outcome 5 (CO5):

4. Explain different amendments on IT Act 2000.
5. Discuss Terrorism on cyberspace.
6. Explain offences of misrepresentation.

Model Question paper

Part A

21. Categorize various patent applications.
22. Explain the criteria for categorizing an invention as patentable or non-patentable.
23. What are the requirements for filing trademarks?
24. Explain copyright and the rights conferred by copyrights.
25. Explain the term geographical indications by giving suitable examples.
26. What is meant by design under the Design Act,2000?
27. Describe the risks associated with cyber space.
28. What is meant by the term cyber laws.
29. Explain cyber stalking and phishing.
30. Define the term hacking and explain its essentials. [3 x 10 =30 Marks]

Part B

Module 1

31. Describe the procedure for registration of patents. [6 Marks]

OR

32. Write short notes on
    c. Intellectual property and the need for its protection. [3 marks]
    d. Importance and features of WIPO. [3 marks]

Module 2

33. Explain the methods for transferring copyrights. [6 Marks]

OR

34. Describe software copyright and how can software be classified according to copyrights. [6 Marks]

Module 3

35. What is industrial design? Describe the salient features of Design act, 2000. [6 Marks]

OR

36. How are the trade secrets dealt with under the Indian law? Discuss. [6 Marks]

Module 4

37. Explain the essential requirements of cyber squatting. [6 Marks]

OR

38. Discuss about cyber space and the protection of copyrights on cyberspace. [6 Marks]

Module 5

39. Explain the objectives and features of Information Technology Act 2000. [6 Marks]

OR

40. What do you mean by cyber crimes? Discuss the nature and types of cyber crimes. [6 Marks]

Syllabus

Module	**Contents**	**Hours**
I	Fundamentals of IPR- Introduction – Intellectual property – Need for protection of intellectual property – WIPO – Intellectual property rights and development – Rationale of protection – TRIPS Agreement - **Patents : –** Introduction **–** Patentable and Non-patentable Invention – Types of patent applications – Guidelines for registration of patent – patent filing – grant of patent – types of patent documents.	10

Module	Contents	Hours
II	**Trademarks**– Introduction – Guidelines for registration-Requirements for filing trademarks – Trademark Infringement – Protection of trademarks – **Copyright –** Introduction – Rights conferred by copyright – registration – ownerships – terms – transfer of copyrights – copyright infringement – databases and copyright-**Software Copyright**–Introduction – Need of software copyright – classification of software according to copyright – software auditing –copyright notice – transfer of copyright.	10
III	**Industrial Designs**– Introduction – Need for protection of design – requirements for registration of designs – Design Act,2000 – Duration of registration of design – application procedure – **GeographicIndications –**Introduction – Filing – **Granting** – Protection of geographic indications**.** **Trade Secret –** definition – discovering and protecting of trade secret.	10
IV	**Cyber law** - Need for cyber laws - Historical perspective - cyberspace - deception by squatting in cyberspace - protection of copyright on cyberspace - infringement of copyright on cyberspace - linking,hyperlinking and framing - ISP in cyberspace - cyberspace and protection of patents in India.	8
V	**Information Technology Act and Punishments-** Introduction to IT Act 2000- Amendments on IT Act - Violation of the right of privacy in cyberspace/internet-punishment for violation of privacy, breach of confidentiality and privacy under IT act-Terrorism on cyberspace Overview of cybercrimes-offences by intermediaries- offences related to protected system- offences of misrepresentation-punishment for Abetment and Attempt to commit offences under the IT act.	10

Textbooks:

1.Dr. R. Radhakrishnan and Dr. S. Balasubramanian, "**Intellectual Property Rights: Text and Cases**", Excel Books
2.Harish Chander, "**Cyber Law and IT Protection**", PHI Learning Pvt.Ltd.

Reference Books:

5. D.Bainbridge, "**Introduction to Computer Law**", Pearson Education
6. RohasNagpal, "**Cyber Crime & Corporate Liability**", CCH, 2008
7. https://www.udemy.com/course/cyber-security-law/
8. https://www.coursera.org/specializations/introduction-intellectual-property

Course Contents and Lecture Schedule

No	Topic	No. of Lectures
1	**Introduction**	
1.1	**Fundamentals of IPR-** Introduction – Intellectual property – Need for protection of intellectual property	2
1.2	WIPO – Intellectual property rights and development – Rationale of protection – TRIPS Agreement	3
1.3	**Patents :** – Introduction – Patentable and Non-patentable Invention – Types of patent applications – Guidelines for registration of patent – patent filing – grant of patent – types of patent documents	5
2		
2.1	**Trademarks**– Introduction – Guidelines for registration – Requirements for filing trademarks – Trademark Infringement – Protection of trademarks	3
2.2	**Copyright –** Introduction – Rights conferred by copyright – registration – ownerships – terms – transfer of copyrights – copyright infringement – databases and copyright	3
2.3	**Software Copyright**– Introduction – Need of software copyright – classification of software according to copyright – software auditing – copyright notice – transfer of copyright.	4
3		
3.1	**Industrial Designs**– Introduction – Need for protection of design – requirements for registration of designs – Design Act,2000 – Duration of registration of design – application procedure	4

No.	**Topic**	**No. of Lectures**
3.2	**Geographic Indications** – Introduction – Filing **Granting** – Protection of geographic indications**.**	4
3.3	**Trade Secret** – definition – discovering and protecting of trade secret.	2
4		
4.1	**Cyber law** - Need for cyber laws - Historical perspective - cyberspace - deception by squatting in cyberspace.	3
4.2	Protection of copyright on cyberspace - infringement of copyright on cyberspace - linking, hyper linking and framing -	3
4.3	ISP in cyberspace - cyberspace and protection of patents in India.	2
5		
5.1	**Information Technology Act and Punishments**- Introduction to IT Act2000- Amendments on IT Act	2
5.2	Violation of the right of privacy in cyberspace/internet-punishment for violation of privacy, breach of confidentiality and privacy under IT act-Terrorism on cyberspace overview of cybercrimes	4
5.3	Offences by intermediaries- offences related to protected system- offences of misrepresentation-punishment for Abetment and Attempt to commit offences under the IT act.	4

20MCA132	OBJECT ORIENTED PROGRAMMING LAB	CATEGORY	L	T	P	CREDIT
		PRACTICAL	0	1	3	2

Preamble: This course enables the students to understand the concepts of object-oriented programming and to develop skills using these paradigms using Java.

Prerequisite: Knowledge of any programming language preferred.

Course Outcomes: After the completion of the course the student will be able to

CO 1	Understand object-oriented concepts and design classes and objects to solve problems
CO 2	Implement arrays and strings.
CO 3	Implement object-oriented concepts like inheritance, overloading and interfaces
CO 4	Implement packages, exception handling, multithreading and generic programming. Use java.util package and Collection framework
CO 5	Develop applications to handle events using applets
CO 6	Develop applications using files and networking concepts

Mapping of course outcomes with program outcomes

	PO 1	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO 10**	**PO 11**	**PO 12**
CO 1	2	2	2	2	3							
CO 2	**3**	2	2		3							
CO 3	**3**	2	2		3							
CO 4	**3**	2	2		3							
CO 5	**3**	3	3		3	2			3		3	
CO 6	**3**	3	3		3	2			3		3	

Assessment Pattern

Bloom's Category	**Continuous Assessment Tests**		**End Semester Examination**
	1	**2**	
Remember(K1)			
Understand(K2)			
Apply(K3)	10	10	10
Analyse(K4)	10	10	10
Evaluate(K5)	10	10	10
Create(K6)	20	20	20

Mark distribution

Total Marks	CIE	ESE	ESE Duration
100	50	50	3 hours

Continuous Internal Evaluation Pattern:

Maximum Marks: 50	
Attendance	15%
Maintenance of daily lab record and GitHub management	20%
Regular class viva	15%
Timely completion of day to day tasks	20%
Tests/Evaluation	30%

End Semester Examination Pattern:

Maximum Marks: 50			
Verification of Daily program record and Git Repository			5 marks
Viva			10 marks
Problem solving (Based on difficulty level, one or more questions may be given)	Flowchart / Algorithm / Structured description of problem to explain how the problem can be solved / Interface Design	15%	35 marks
	Program correctness	50%	
	Code efficiency	15%	
	Formatted output and Pushing to remote Git repository	20%	
Total Marks			50 marks

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Define a class ‘product’ with data members pcode, pname and price. Create 3 objects of the class and find the product having the lowest price.
2. Read 2 matrices from the console and perform matrix addition.

3. Add complex numbers
4. Read a matrix from the console and check whether it is symmetric or not.
5. Create CPU with attribute price. Create inner class Processor (no. of cores, manufacturer) and static nested class RAM (memory, manufacturer). Create an object of CPU and print information of Processor and RAM.

Course Outcome 2 (CO2)

1. Program to Sort strings
2. Search an element in an array.
3. Perform string manipulations
4. Program to create a class for Employee having attributes eNo, eName eSalary. Read n employ information and Search for an employee given eNo, using the concept of Array of Objects.

Course Outcome 3(CO3):

1. Area of different shapes using overloaded functions
2. Create a class 'Employee' with data members Empid, Name, Salary, Address and constructors to initialize the data members. Create another class 'Teacher' that inherit the properties of class employee and contain its own data members department, Subjects taught and constructors to initialize these data members and also include display function to display all the data members. Use array of objects to display details of N teachers.
3. Create a class 'Person' with data members Name, Gender, Address, Age and a constructor to initialize the data members and another class 'Employee' that inherits the properties of class Person and also contains its own data members like Empid, Company_name, Qualification, Salary and its own constructor. Create another class 'Teacher' that inherits the properties of class Employee and contains its own data members like Subject, Department, Teacherid and also contain constructors and methods to display the data members. Use array of objects to display details of N teachers.
4. Write a program has class Publisher, Book, Literature and Fiction. Read the information and print the details of books from either the category, using inheritance.
5. Create classes Student and Sports. Create another class Result inherited from Student and Sports. Display the academic and sports score of a student.

6. Create an interface having prototypes of functions area() and perimeter(). Create two classes Circle and Rectangle which implements the above interface. Create a menu driven program to find area and perimeter of objects.

7. Prepare bill with the given format using calculate method from interface.

Order No.

Date :

Product Id	Name	Quantity	unit price	Total
101	A	2	25	50
102	B	1	100	100
			Net. Amount	150

Course Outcome 4 (CO4):

1. Create a Graphics package that has classes and interfaces for figures Rectangle, Triangle, Square and Circle. Test the package by finding the area of these figures.
2. Create an Arithmetic package that has classes and interfaces for the 4 basic arithmetic operations. Test the package by implementing all operations on two given numbers
3. Write a user defined exception class to authenticate the user name and password.
4. Find the average of N positive integers, raising a user defined exception for each negative input.
5. Define 2 classes; one for generating multiplication table of 5 and other for displaying first N prime numbers. Implement using threads. (Thread class)
6. Define 2 classes; one for generating Fibonacci numbers and other for displaying even numbers in a given range. Implement using threads. (Runnable Interface)
7. Producer/Consumer using ITC
8. Program to create a generic stack and do the Push and Pop operations.
9. Using generic method perform Bubble sort.
10. Maintain a list of Strings using ArrayList from collection framework, perform built-in operations.
11. Program to remove all the elements from a linked list
12. Program to remove an object from the Stack when the position is passed as parameter
13. Program to demonstrate the creation of queue object using the PriorityQueue class
14. Program to demonstrate the addition and deletion of elements in deque
15. Program to demonstrate the creation of Set object using the LinkedHashset class
16. Write a Java program to compare two hash set

17. Program to demonstrate the working of Map interface by adding, changing and removing elements.
18. Program to Convert HashMap to TreeMap

Course Outcome 5 (CO5):

1. Program to draw Circle, Rectangle, Line in Applet.
2. Program to find maximum of three numbers using AWT.
3. Find the percentage of marks obtained by a student in 5 subjects. Display a happy face if he secures above 50% or a sad face if otherwise.
4. Using 2D graphics commands in an Applet, construct a house. On mouse click event, change the color of the door from blue to red.
5. Implement a simple calculator using AWT components.
6. Develop a program that has a Choice component which contains the names of shapes such as rectangle, triangle, square and circle. Draw the corresponding shapes for given parameters as per user's choice.
7. Develop a program to handle all mouse events and window events
8. Develop a program to handle Key events.

Course Outcome 6 (CO6):

1. Program to list the sub directories and files in a given directory and also search for a file name.
2. Write a program to write to a file, then read from the file and display the contents on the console.
3. Write a program to copy one file to another.
4. Write a program that reads from a file having integers. Copy even numbers and odd numbers to separate files.
5. Client server communication using Socket – TCP/IP
6. Client Server communication using DatagramSocket - UDP

Syllabus:

Classes and Objects, Constructors, Method Overloading, Access Modifiers, Arrays and Strings, Inheritance, Interfaces, Abstract classes, Dynamic Method Dispatch, String, Packages, Introduction to java.util, Collection framework, User defined packages, Exceptions, Multithreading, Applets, Graphics, File, Generic programming, Socket Programming

Reference Books

1. Herbert Schildt, “***Java The Complete Reference***”, Seventh Edition, Tata McGraw-Hill Edition

2. C. Thomas Wu, “***An introduction to Object-oriented programming with Java***”, Fourth Edition, Tata McGraw-Hill Publishing company Ltd.

3. Cay S. Horstmann and Gary Cornell, “***Core Java: Volume I – Fundamentals***”, Eighth Edition, Sun Microsystems Press.

4. K. Arnold and J. Gosling, “***The JAVA programming language***”, Third edition, Pearson Education.

5. Paul Deitel and Harvey Deitel, “***Java, How to Program***”, Tenth Edition, Pearson Education

6. Rohit Khurana, “***Programming with Java***”, Vikas Publishing, 2014.

7. Timothy Budd, “***Understanding Object-oriented programming with Java***”, Updated Edition, Pearson Education.

8. Y. Daniel Liang, “***Introduction to Java programming***”, Seventh Edition, Pearson Education.

Web Reference

- https://www.hackerrank.com/domains/java
- https://www.geeksforgeeks.org/java-tutorial/
- https://www.w3resource.com/java-tutorial/
- https://www.w3resource.com/java-exercises/
- https://nptel.ac.in/courses/106/105/106105191/
- https://nptel.ac.in/noc/courses/noc20/SEM1/noc20-cs08/
- https://www.coursera.org/learn/object-oriented-java
- https://www.edx.org/course/object-oriented-programming-in-java-2

Course Contents and Lab Schedule

Topic	No. of hours
1. Classes and Objects.	3
2. Constructors, Method Overloading, Access Modifiers	2
3. Arrays and Strings.	4
4. Inner class – static and non-static	2
5. Inheritance, Multiple inheritance - implementation using interfaces	3
6. Method overriding, Abstract classes, Dynamic Method Dispatch	3
7. Interfaces and Packages, StringBuffer class	3
8. Introduction to java.util package – Vector, Scanner, StringTokenizer	2
9. Collection framework – ArrayList, LinkedList, Stack, Queue, Set, Map	3
10. User defined packages	2
11. Exceptions – User defines exceptions	2
12. Multithreading – Thread class	2
13. Inter Thread Communication	2
14. Generic programming	2
15. Applets, Graphics – 2D	3
16. Event handling in Applet	3
17. File	3
18. Socket Programming	3

20MCA134	ADVANCED DBMS LAB	CATEGORY	L	T	P	CREDIT
		PRACTICAL	0	1	3	2

Preamble: This course is to provide understanding on relational and non-relational database systems and its design. The course covers SQL, PL/SQL and NoSQL programs which are essential for the development and deployment of web based applications. Also this course serves as a prerequisite for many advanced courses in Data Science areas.

Prerequisite: Database Management Systems

Course Outcomes: After the completion of the course the student will be able to

CO 1	Design and build a simple relational database system and demonstrate competence with the fundamentals tasks involved with modelling, designing and implementing a database.
CO 2	Apply PL/SQL for processing databases.
CO 3	Comparison between relational and non-relational (NoSQL) databases and the configuration of NoSQL Databases.
CO 4	Apply CRUD operations and retrieve data in a NoSQL environment.
CO 5	Understand the basic storage architecture of distributed file systems.
CO 6	Design and deployment of NoSQL databases with real time requirements.

Mapping of course outcomes with program outcomes

	PO 1	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO 10**	**PO 11**	**PO 12**
CO 1	3	2	3	2	2					1	1	
CO 2	2	2	2		1							
CO 3	2	2	2	2						1	1	
CO 4	2	2	3	1	2		1			1	1	1
CO 5	3	2	2				1				1	1
CO 6	2	2	3	1	1			1		1	1	2

Assessment Pattern

Bloom's Category	**Continuous Assessment Tests**		**End Semester Examination**
	1	**2**	
Remember(K1)			
Understand(K2)			
Apply(K3)	10	10	10
Analyse(K4)	10	10	10
Evaluate(K5)	10	10	10
Create(K6)	20	20	20

Mark distribution

Total Marks	**CIE**	**ESE**	**ESE Duration**
100	50	50	3 hours

Continuous Internal Evaluation Pattern:

Maximum Marks: 50	
Attendance	15%
Maintenance of daily lab record and GitHub management	20%
Regular class viva	15%
Timely completion of day to day tasks	20%
Tests/Evaluation	30%

End Semester Examination Pattern:

Maximum Marks: 50			
Verification of Daily program record and Git Repository			5 marks
Viva			10 marks
Problem solving (Based on difficulty level, one or more questions may be given)	Flowchart / Algorithm / Structured description of problem to explain how the problem can be solved / Interface Design	15%	35 marks
	Program correctness	50%	
	Code efficiency	15%	
	Formatted output and Pushing to remote Git repository	20%	
Total Marks			50 marks

Course Level Assessment

Questions Course Outcome 1 (CO1):

1. Creation of a database using DDL commands including integrity constraints. (K6)
2. Create an application to apply Data Manipulation Language (DML) commands to modify the database. (K6)
3. Apply DCL and TCL commands to impose restrictions on databases. (K3)
4. Create an application to retrieve data from databases using select, views. (K6)
5. Create an application to use joins for query optimization. (K6)

Course Outcome 2 (CO2):

1. Construct PL/SQL code for sample databases. (K6)

Course Outcome 3(CO3):

1. Compare relational and non-relational databases. (K5)
2. Understand the installation and configuration of NoSQL Databases. (K2)

Course Outcome 4 (CO4):

1. Build sample collections/documents to perform query operations. (K6)

Course Outcome 5 (CO5):

1. Build sample collections/documents to perform the shell commands like replica set, indexing etc. (K6)

Course Outcome 6 (CO6):

1. Develop sample applications using any of the front end tools and NoSQL. (K6)
2. Usage of concerned Online/Cloud Storage Management Systems like MongoDB Atlas, Cassandra DataStax etc. (K6)
3. Deployment of NoSQL in Cloud: Google Bigtable/ Amazon DynamoDB/ Azure Cosmos DB. (K6**)**

Syllabus

1. An overview of relational database design using MySQL/ MariaDB/ PostgreSQL etc. (Apply the following basic queries on an Employee/ Student database etc.)
    a. DDL Commands
    b. DML Commands
    c. Imposing restrictions on database (DCL & TCL Commands)
    d. Accessing database (SELECT, Filtering using WHERE, HAVING, GROUP BY, ORDER BY Clauses, Subquery and View)
    e. Optimizing databases (Join, Aggregate & Set operations, Other operators like arithmetic, logical, special etc.)
2. PL/SQL Programs (Trigger, Cursor, Stored Procedures and Functions)
3. Introduction to NoSQL Databases.
    a. Installation and configuration of any one of the NoSQL databases - MongoDB/ Cassandra/ HBase/ CouchDB/ Amazon DynamoDB/ Redis/ Neo4j etc.
4. Designing Databases using NoSQL
5. Query Processing
    a. Performing CRUD operations
    b. Retrieving Data from a NoSQL database
    c. Usage of aggregate functions, regular expressions etc.
6. NoSQL Administration
    a. Security, Monitoring & Backup
    b. Create Users and Roles
7. NoSQL shell commands
    a. Perform Sharding, Replication (Master-Slave/ Master-Less/ Peer-to-Peer Architectures), Clustering, Partitioning, Indexing (Corresponding to the selected NoSQL Database)

8. Deployment
    a. Local Deployment
        i. NoSQL and Front-End: PHP/Java/Python (MongoDB/ Cassandra etc.)
    b. Cloud Deployment
        i. NoSQL and Cloud: Amazon DynamoDB/ Google Bigtable/ Azure Cosmos DB
        ii. Familiarization of Atlas/ DataStax corresponding to the selected NoSQL Database
9. ***Micro project:*** *Students can be given a group micro project, so that they learn to work in a team environment.*

Text Books

1. Abraham Silberschatz, Henry F. Korth, S. Sudarshan," ***Database System Concepts***", McGraw Hill Education, 6th Edition (2011)
2. Guy Harrison, "***Next Generation Databases: NoSQL, NewSQL, and Big Data***", Apress, 1st Edition (14 December 2015)

Reference Books

1. Raghu Ramakrishnan and Johannes Gehrke, "***Database Management Systems***", McGraw Hill, 3rd Edition (2014).
2. HBase: The Definitive Guide. Lars George O'Reilly Media; August 2011, ISBN: 9781449315771
3. Shashank Tiwari. Professional NoSQL. John Wiley and Sons. ISBN: 978-0-470-94224-6.
4. MongoDB Administrator's Guide, Cyrus Dasadia, October 2017, Packet Publishing ISBN: 9781787126480
5. Cassandra: The Definitive Guide Distributed Data at Web Scale, 1st Edition, Eben Hewitt, Jeff Carpenter, O'Reilly Media; November 2010

Web Resources

1. Database Management System https://nptel.ac.in/courses/106/105/106105175/
2. Databases: SQL https://www.edx.org/course/databases-5-sql
3. Introduction to MongoDB https://www.coursera.org/learn/introduction-mongodb
4. Apache Cassandra https://www.edureka.co/cassandra
5. NoSQL systems https://www.coursera.org/learn/nosql-databases
6. https://hbase.apache.org/
7. https://couchdb.apache.org/ https://aws.amazon.com/dynamodb/
8. https://aws.amazon.com/dynamodb/

Course Contents and Lecture Schedule

No	Topic	No. of Lectures
1	*An overview of relational database design using MySQL/ MariaDB/ PostgreSQL etc. (Apply the following basic queries on an Employee/ Student database etc.)*	*6 hrs*
1.1	• DDL Commands • DML Commands • Imposing restrictions on database (DCL & TCL Commands)	3
1.2	• Accessing database (SELECT, Filtering using WHERE, HAVING, GROUP BY, ORDER BY Clauses, Subquery and View) • Optimizing databases (Join, Aggregate & Set operations, Other operators like arithmetic, logical, special etc.)	3

No	*Topic*	*No. of Lectures*
2	*PL/SQL Programs*	*4 hrs*
2.1	• Trigger, Cursor, Stored Procedures and Functions	4
3	*Introduction to NoSQL Databases*	*2 hrs*
3.1	• Installation and configuration of any one of the NoSQL databases - MongoDB/ Cassandra/ HBase/ CouchDB/ Amazon DynamoDB/ Redis/ Neo4j etc.	2
4	*Designing Databases using NoSQL*	*2 hrs*
5	*Query Processing*	*8 hrs*
5.1	• Performing CRUD operations • Retrieving Data from a NoSQL database • Usage of aggregate functions, regular expressions etc.	8
6	*NoSQL Administration*	*2 hrs*
6.1	• Security, Monitoring & Backup • Create Users and Roles	2
7	*NoSQL shell commands*	*6 hrs*
7.1	• Perform Sharding, Replication (Master-Slave/ Master-Less/ Peer-to-Peer Architectures), Clustering, Partitioning, Indexing (Corresponding to the selected NoSQL Database)	6
8	*Deployment*	*16 hrs*
8.1	• Local Deployment NoSQL and Front-End: PHP/Java/Python (MongoDB/ Cassandra etc.)	4
8.2	• Cloud Deployment NoSQL and Cloud: Amazon DynamoDB/ Google Bigtable/ Azure Cosmos DB	8
8.3	• Familiarization of Atlas/ DataStax corresponding to the selected NoSQL Database	4
9	*Micro project*	*10 hrs*

20MCA136	**NETWORKING & SYSTEM ADMINISTRATION LAB**	**CATEGORY**	**L**	**T**	**P**	**CREDIT**
		PRACTICAL	**0**	**1**	**3**	**2**

Preamble: This laboratory course is intended to provide the background knowledge required for a software professional in the fields of networking and system administration. Students will acquire necessary knowledge to deploy and administer systems.

Prerequisite: Basic understanding of computer programming, Internet and operating systems

Course Outcomes: After the completion of the course the student will be able to

CO 1	Install and configure common operating systems.
CO 2	Perform system administration tasks.
CO 3	Install and manage servers for web applications.
CO 4	Write shell scripts required for system administration.
CO 5	Acquire skill sets required for a DevOps.

Mapping of course outcomes with program outcomes

	PO 1	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO 10**	**PO 11**	**PO 12**
CO 1	1		2		2							
CO 2	1		2									
CO 3			2		2							
CO 4					2							
CO 5	2				2							

Assessment Pattern

Bloom's Category	**Continuous Assessment Tests**		**End Semester Examination**
	1	**2**	
Remember(K1)			
Understand(K2)			
Apply(K3)	10	10	10
Analyse(K4)	10	10	10
Evaluate(K5)	10	10	10
Create(K6)	20	20	20

Mark distribution

Total Marks	CIE	ESE	ESE Duration
100	50	50	3 hours

Continuous Internal Evaluation Pattern:

Maximum Marks: 50	
Attendance	15%
Maintenance of daily lab record and GitHub management	20%
Regular class viva	15%
Timely completion of day to day tasks	20%
Tests/Evaluation	30%

End Semester Examination Pattern:

<table>
<tr><th colspan="4">Maximum Marks: 50</th></tr>
<tr><td colspan="3">Verification of Daily program record and Git Repository</td><td>5 marks</td></tr>
<tr><td colspan="3">Viva</td><td>10 marks</td></tr>
<tr><td rowspan="4">Problem solving (Based on difficulty level, one or more questions may be given)</td><td>Flowchart / Algorithm / Structured description of problem to explain how the problem can be solved / Interface Design</td><td>15%</td><td rowspan="4">35 marks</td></tr>
<tr><td>Program correctness</td><td>50%</td></tr>
<tr><td>Code efficiency</td><td>15%</td></tr>
<tr><td>Formatted output and Pushing to remote Git repository</td><td>20%</td></tr>
<tr><td colspan="3">Total Marks</td><td>50 marks</td></tr>
</table>

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Install latest version of Ubuntu on a virtual box, set up a static ip address to it and install drupal environment.
2. You are given a computer with very low hardware resources. It is to be used as a kiosk. Identify and install a suitable Linux distribution. You can simulate it in a virtual environment.

Course Outcome 2 (CO2)

1. You are given a system which is connected to internet. However, users logging on to the system are unable to access internet from their browser. Trouble shoot the issue,clearly documenting the steps you have taken (Possible issues to look for are browser configuration, network connectivity, routing, ip address configuration,DNS resolution)
2. You are given a system which boots to a non graphical environment. You are also given a shell script which is designed for a specific task. Your task is to make sure that the script runs every time the system boots up. Write/modify necessary scripts for this.
3. You are required to add 100 users to a Linux system. Details of the users to be added were collected from a web form to a csv file. The csv may contain errors such as wrong case or missing fields. Write a script to add users using the data provided in the csv file with proper error checking.

Course Outcome 3(CO3):

1. You are given a bare bone installation of latest version Ubuntu. Assume that the system is accessible from internet. Your task is to successfully install word press (or any other web application) on this server. Clearly indicate the steps taken and software installed for this task.

2. Assume that you have an installation of old version Ubuntu. However, it does not have the latest version of virtual box (or some other application). The new version is available as a binary on a website. Upgrade to this version.

Course Outcome 4 (CO4):

1.Look at the system log files. Write a shell script to extract the last login details of a particular user and list out all failed logins. Store the results to a file. The user name should be given as a command line argument.

2.Write a shell script to display the details of a particular process currently running. Assume that you have necessary permissions. The process name/id is to be given as a command line argument

Course Outcome 5 (CO5):

1.Capture network traffic on your system. Using wireshark find out all http and https traffic to a specific host.

2.Write an Ansible playbook to deploy a new Linux VM on a remote server.

Syllabus:

Introduction to Computer hardware. Study of various peripherals. Study of common operating systems. File system organization in common operating systems.

Study of command line environment in common operating systems. Study of command line tools for system administration.

Shell scripting: bash shell, shell scripts for system management.

Study of startup scripts.

Study of server software for common applications such as http, ftp, dns, dhcp.

Practical study of Ipv4 and Ipv6 networking protocols. Setting up firewalls.

Virtual machines and containers. Configuration and deployment.

List of Lab Experiments/Exercises

To gain proficiency in command line tools and operations, it is highly recommended to use a terminal window instead of GUI tools. This will later help the student with latest approaches in maintaining cloud based infrastructure. virtualbox/quemu. may be used for this.

1. Introduction to Computer hardware: Physical identification of major components of a computer system such as mother board, RAM modules, daughter cards, bus slots, SMPS, internal storage devices,i nterfacing ports. Specifications of desktop and server class computers. Installation of common operating systems for desktop and server use. (Students may be asked to formulate specification for computer to be used as Desktop, Web server)

2. Study of a terminal based text editor such as Vim or Emacs. (By the end of the course, students are expected to acquire following skills in using the editor: cursor operations, manipulate text, search for patterns, global search and replace)
   Basic Linux commands, familiarity with following commands/operations expected
   1. man
   2. ls, echo, read
   3. more, less, cat,
   4. cd, mkdir, pwd, find
   5. mv, cp, rm ,tar
   6. wc, cut, paste
   7. head, tail, grep, expr
   8 chmod, chown
   9. Redirections & Piping
   10. useradd, usermod, userdel, passwd
   11. df,top, ps
   12 ssh, scp, ssh-keygen, ssh-copy-id

   .

3. File system hierarchy in a common Linux distribution, file and device permissions, study of system configuration files in /etc, familiarizing log files for system events, user activity, network events.

4. Shell scripting: study bash syntax, environment variables, variables, control constructs such as if, for and while, aliases and functions, accessing command line arguments passed to shell

scripts. Study of startup scripts, login and logout scripts, familiarity with systemd and system 5 init scripts is expected.

5. Installation and configuration of LAMP stack. Deploy an open source application such as phpmyadmin and Wordpress.

6. Installation and configuration of common software frame works such as Laravel. (Student should acquire the capability to install and configure a modern framework)

7. Build and install software from source code, familiarity with make and cmake utilities expected.

8. Introduction to command line tools for networking
IPv4 networking, network commands: ping route traceroute, nslookup, ip. Setting up static and dynamic IP addresses. Concept of Subnets, CIDR address schemes, Subnet masks, iptables, setting up a firewall for LAN, Application layer (L7) proxies.

9. Analyzing network packet stream using tcpdump and wireshark. Perform basic network service tests using nc.

10. Introduction to Hypervisors and VMs, Xen or KVM , Introduction to Containers: Docker, installation and deployment.

11. Automation using Ansible: Spin up a new Linux VM using Ansible playbook

SEMESTER -3

CODE	COURSE NAME	CATEGORY	L	T	P	CREDIT
20MCA201	DATA SCIENCE & MACHINE LEARNING	CORE	3	1	0	4

Preamble: This is an introductory course on data science and basic concepts behind various machine learning techniques. Machine learning is the study of adaptive computational systems that improve their performance with experience. At the end of the course the students should be able to design and implement machine learning solutions to classification, regression, and clustering problems and to evaluate and interpret the results of the algorithms.

Prerequisite: Probability and Statistics, Linear Algebra, Programming in Python/R.

Course Outcomes: After the completion of the course the student will be able to:

CO No.	Course Outcome (CO)	Bloom's Category Level
CO 1	Discuss the fundamental concepts of data science and data visualization techniques.	Level 2: Understand
CO 2	Explain the basics of machine learning and use lazy learning and probabilistic learning algorithms to solve data science problems.	Level 3: Apply
CO 3	Describe decision trees, classification rules & regression methods and how these algorithms can be applied to solve data science problems.	Level 3: Apply
CO 4	Solve data science problems using neural networks and support vector machines.	Level 3: Apply
CO 5	Discuss clustering using k-means algorithm and evaluate & improve the performance of machine learning classification models.	Level 3: Apply

Mapping of Course Outcomes with Program Outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	1	1	-	-	-	-	3	-	-	-	-	-
CO 2	3	3	3	2	-	-	3	-	-	-	-	-
CO 3	3	3	3	2	-	-	3	-	-	-	-	-
CO 4	3	3	3	2	-	-	3	-	-	-	-	-
CO 5	3	3	3	2	-	-	3	-	-	-	-	-

3/2/1: High/Medium/Low

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination
	1	2	
Remember (K1)	15	10	10
Understand (K2)	25	20	30
Apply (K3)	10	20	20
Analyse (K4)			
Evaluate (K5)			
Create (K6)			

Mark Distribution

Total Marks	CIE	ESE	ESE Duration
100	40	60	3 hours

Continuous Internal Evaluation Pattern:

Attendance	: 8 marks
Continuous Assessment Test (2 numbers)	: 20 marks
Assignment/Quiz/Course project	: 12 marks

End Semester Examination Pattern: There will be two parts: Part A and Part B. Part A contains 10 compulsory short answer questions, 2 from each module. Each question carries 3 marks. Part B contains 2 questions from each module, of which the student should answer any one. Each question can have a maximum of 2 subdivisions and carry 6 marks.

Sample Course Level Assessment Questions

Course Outcome 1 (CO1):

1. What is data science and why do we need data science?
2. Explain the data science classification and illustrate data science tasks.
3. Describe the various methods to understand data.
4. Explain the typical methods to visualize data.

Course Outcome 2 (CO2)

1. Explain the differences between supervised and unsupervised machine learning algorithms.
2. Describe the key concepts that define nearest neighbour classifiers, and why they are considered "lazy" learners.
3. Explain how to apply *k*-NN classifier in a data science problem.
4. State Bayes' theorem in statistics. Outline the Naive Bayes algorithm to build classification models.
5. Use Naive Bayes algorithm to determine whether a red domestic SUV car is a stolen car or not using the following data:

Example	**Colour**	**Type**	**Origin**	**Stolen?**
1	red	sports	domestic	yes
2	red	sports	domestic	no
3	red	sports	domestic	yes
4	yellow	sports	domestic	no
5	yellow	sports	imported	yes
6	yellow	SUV	imported	no
7	yellow	SUV	imported	yes
8	yellow	SUV	domestic	no
9	red	SUV	imported	no
10	red	sports	imported	yes

Course Outcome 3 (CO3):

1. Classify data science tasks using decision trees and classification rule learners.
2. Discuss the various feature selection measures.
3. How to simplify a decision tree by pruning.
4. Describe how to construct classification rules from decision trees.
5. Explain the concepts of regression and correlation.
6. How to estimate a linear regression model.
7. Consider the following set of training examples:

Instance	**Classification**	**a_1**	**a_2**
1	+	T	T
2	+	T	T

3	-	T	F
4	+	F	F
5	-	F	T
6	-	F	T

a) Find the entropy of this collection of training examples with respect to the target function “classification”?

b) Calculate the information gain of **a$_2$** relative to these training examples?

Course Outcome 4 (CO4):

1. Explain how artificial neural networks mimic human brain to model arbitrary functions and how these can be applied to real-world problems.

2. Describe different activation functions and network topology.

3. Discuss basic idea behind the backpropagation algorithm.

4. Explain how a support vector machine can be used for classification of linearly separable data.

5. How to compute the distance of a point from a hyperplane.

6. How the kernel trick is used to construct classifiers in nonlinearly separated data.

Course Outcome 5 (CO5):

1. Explain how the clustering tasks differ from the classification tasks.

2. How clustering defines a group, and how such groups are identified by *k*-means clustering algorithm.

3. Find the three clusters after one epoch for the following eight examples using the *k*-means algorithm and Euclidean distance: A1 = (2,10), A2 = (2,5), A3 = (8,4), A4 = (5,8), A5 = (7,5), A6 = (6,4), A7 = (1,2), A8 = (4,9). Suppose that the initial seeds (centres of each cluster) are A1, A4 and A7.

4. Explain the various matrices used to measure the performance of classification algorithms

5. Explain the concepts of bagging and boosting.

6. Suppose 10000 patients get tested for flu; out of them, 9000 are actually healthy and 1000 are actually sick. For the sick people, a test was positive for 620 and negative for 380. For the healthy people, the same test was positive for 180 and negative for 8820. Construct a confusion matrix for the data and compute the precision and recall for the data.

Model Question Paper
Course Code: 20MCA201

Course Name: DATA SCIENCE AND MACHINE LEARNING

Max. Marks :60 **Duration: 3 Hrs**

Part A

Answer all questions. Each question carries 3 marks (10 * 3 = 30 Marks)

1. What is data science?
2. Explain the different types of data.
3. Differentiate between supervised and unsupervised learning algorithms.
4. Explain how to choose the value of *k* in *k*-NN algorithm.
5. Explain entropy and information gain.
6. Explain the Ordinary Least Square method in regression.
7. Define activation function. Give two examples.
8. What is maximum margin hyperplane.
9. Define precision, recall and F-measure.
10. Explain bootstrap sampling

Part B

Answer one full question from each module, each carries 6 marks.

11. Explain the various methods for visualising multivariate data. (6 marks)

OR

12. Explain the various processes for preparing a dataset to perform a data science task. (6 marks)

13. Based on a survey conducted in an institution, students are classified based on the two attributes of academic excellence and other activities. Given the following data, identify the classification of a student with X = 5 and Y = 7 using *k*-NN algorithm (choose *k* as 3).

X (Academic Excellence)	Y (Other Activities)	Z (Classification)
8	6	Outstanding
5	6	Good
7	3	Good
6	9	Outstanding

(6 marks)

OR

14. Given the following data on a certain set of patients seen by a doctor. Can the doctor conclude that a person having chills, fever, mild headache and without running nose has flu? (Use Naive Bayes classification).

Chills	Running nose	Headache	Fever	Has flu
Y	N	mild	Y	N
Y	Y	no	N	Y
Y	N	strong	Y	Y
N	Y	mild	Y	Y
N	N	no	N	N
N	Y	strong	Y	Y
N	Y	strong	N	N
Y	Y	mild	Y	Y

(6 marks)

15. Obtain a linear regression for the data given in the table below assuming that *y* is the independent variable.

x	55	60	65	70	80
y	52	54	56	58	62

(6 marks)

OR

16. Given the following data, draw a decision tree to predict whether a person cheats. Give the corresponding set of classification rules also.

Sl. No.	Refund	Marital status	Income	Cheats?
1	Yes	Single	High	No
2	No	Married	High	No
3	No	Single	Low	No
4	Yes	Married	High	No
5	No	Divorced	High	Yes
6	No	Married	Low	No

7	Yes	Divorced	High	No
8	No	Single	High	Yes
9	No	Married	Low	No
10	No	Single	High	Yes

(6 marks)

17. Define an artificial neuron. What are the characteristics of an artificial neural network (ANN)?

(6 marks)

OR

18. a) Define linearly separable dataset. Give an example each of a dataset that is linearly separable and of a dataset that is not linearly separable.

(3 marks)

b) Define kernel function. Explain the kernel trick to construct a classifier for a dataset that is not linearly separable.

(3 marks)

19. Suppose 10000 patients get tested for flu; out of them, 9000 are actually healthy and 1000 are actually sick. For the sick people, a test was positive for 620 and negative for 380. For the healthy people, the same test was positive for 180 and negative for 8820. Construct a confusion matrix for the data and compute the precision and recall for the data. (6 marks)

OR

20. Assume the following: A database contains 80 records on a particular topic of which 55 are relevant to a certain investigation. A search was conducted on that topic and 50 records were retrieved. Of the 50 records retrieved, 40 were relevant. Construct the confusion matrix for the search and calculate the precision and recall scores for the search. (6 marks)

Syllabus

Module 1 (9 Hours)

Introduction to data science, Data science classification, Data science process - Prior knowledge, Data preparation, Modelling, Application, Data exploration - Data sets, Descriptive statistics for univariate and multivariate data

Data visualisation – Histogram, Quartile plot, Distribution chart, Scatter plot, Bubble chart, Density chart

Module 2 (9 Hours)

Introduction to machine learning: How machines learn - Data storage, Abstraction, Generalisation, Evaluation, Machine learning in practice - Types of machine learning algorithms.

Lazy learning: Classification using K-Nearest Neighbour algorithm - Measuring similarity with distance, Choice of k, Preparing data for use with k-NN.

Probabilistic learning: Understanding Naive Bayes - Conditional probability and Bayes theorem, Naive Bayes algorithm for classification, The Laplace estimator, Using numeric features with Naive Bayes.

Module 3 (9 Hours)

Decision tree learning: Concept of decision tree, Divide and conquer approach, C5.0 Decision tree algorithm, Choosing the best split, Pruning the decision tree.

Classification rules learning: Concept of classification rules, Separate and conquer approach, The 1R algorithm, Rules from decision trees.

Regression methods: Concept of regression, Simple linear regression, Ordinary least squares estimation, Correlations, Multiple linear regression.

Module 4 (9 Hours)

Neural network learning: Artificial neurons, Activation functions, Network topology, Training neural networks with backpropagation.

Support vector machines: Hyperplanes, Classification using hyperplanes, Maximum margin hyperplanes in linearly separable data, Using kernels for non-linear spaces.

Module 5 (9 Hours)

Clustering: The k-means clustering algorithm, Using distance to assign and update clusters, Choosing number of clusters.
Evaluating model performance: Confusion matrices, Precision and recall, Sensitivity and specificity, Precision and recall, F-measure, ROC curves, Cross validation - K-fold cross validation, Bootstrap sampling.
Improving model performance - Bagging, Boosting, Random forests.

Text Books

1. Vijay Kotu, Bala Deshpande, Data Science Concepts and Practice, Morgan Kaufmann Publishers 2018 (Module 1)
2. Brett Lantz, Machine Learning with R, Second edition, PackT publishing 2015 (Modules 2 to 5)

Reference Books

1. Michael Steinbach, Pang-Ning Tan, and Vipin Kumar, Introduction to Data Mining, Pearson 2016.
2. Jiawei Han, Micheline Kamber and Jian Pei, Data mining Concepts and techniques, Morgan Kaufmann Publishers 2012
3. Peter Harrington, Machine Learning in action, Dreamtech publishers 2012
4. Dr M Gopal, Applied Machine learning, McGraw Hill Education Private Limited
5. E. Alpayidin, Introduction to Machine Learning, Prentice Hall of India (2005)
6. T. Hastie, RT Ibrashiran and J. Friedman, The Elements of Statistical Learning, Springer 2001
7. Data Science from Scratch: First Principles with Python, Joel Grus, O'Reilly, First edition, 2015
8. Introducing Data Science, Davy Cielen, Arno D. B. Meysman, Mohamed Ali, Manning Publications Co., 1st edition, 2016

Web Resources:

1. https://www.coursera.org/learn/machine-learning
2. https://www.coursera.org/learn/data-scientists-tools

Course Contents and Lecture Schedule

	Topic	**No. of Lectures**
1	**Module 1**	**9 hrs**
1.1	Introduction to data science - What is data science? Why data science?	2 hrs
1.2	Data science classification	1 hr
1.3	Data science process - Prior knowledge, Data preparation, Modelling, Application	2 hrs
1.4	Data exploration- Data sets, Descriptive statistics for univariate and multivariate data	2 hrs
1.5	Data visualization – Histogram, Quartile plot, Distribution chart, Scatter plot, Bubble chart, Density chart	2 hrs
2	**Module 2**	**9 hrs**

2.1	How machines learn – Data storage – Abstraction – Generalisation – Evaluation	1hr
2.2	Machine learning in practice – Types of machine learning algorithms.	1hr
2.3	Classification: Lazy learning - K-Nearest Neighbour algorithm	2 hrs
2.4	Measure of similarity, Choice of k	1 hr
2.5	Preparing data for use with k-NN	1 hr
2.6	Probabilistic Learning: Conditional probability and Bayes theorem.	1 hr
2.7	Naive Bayes algorithm	2 hrs
3	**Module 3**	**9 hrs**
3.1	Concept of decision tree, Divide and conquer approach	1 hr
3.2	C5.0 Decision tree algorithm	1 hr
3.3	Choosing the best split, Pruning the decision tree	2 hrs
3.4	Classification rules learning: Concept of classification rules, Separate and conquer approach	1 hr
3.5	The 1R algorithm, Rules from decision trees	1 hr
3.6	Regression methods: Concept of regression, Correlations	1 hr
3.7	Simple linear regression, Ordinary least squares estimation	1 hr
3.8	Multiple linear regression	1 hr
4	**Module 4**	**9 hrs**
4.1	Understanding neural networks - Artificial neurons	1 hr
4.2	Activation functions, Network topology	2 hrs
4.3	Training neural networks with back propagation	1hr
4.4	Understanding Support Vector Machines, Classification with hyperplane	1 hr
4.5	Linearly separable data, Nonlinearly separable data	1 hr
4.6	Methods to find maximum margin hyperplanes in linearly separable data	1 hr
4.7	Using kernels for non-linear spaces	2 hrs
5	**Module 5**	**9 hrs**
5.1	Understanding Clustering - The k-means clustering algorithm	1 hr

5.2	Using distance to assign and update clusters, Choosing the appropriate number of clusters	1 hr
5.3	Evaluating model performance: Confusion matrices, Precision and recall, Sensitivity and specificity, Precision and recall, F-measure, ROC curves.	2 hrs
5.4	Cross validation: K-fold cross validation, Bootstrap sampling	2 hrs
5.5	Improving model performance: Bagging, Boosting	2 hrs
5.6	Random forests	1 hr

CODE	COURSE NAME	CATEGORY	L	T	P	CREDIT
20MCA203	DESIGN & ANALYSIS OF ALGORITHMS	CORE	3	1	0	4

Preamble: The syllabus is prepared with a view to provide a strong foundation to students in design and analysis of computer algorithms and to introduce them the advanced topics such as Network Flows, Approximation algorithms and Randomised algorithms.

Prerequisite: Knowledge in Data Structures

Course Outcomes: After completion of the course the student will be able to

CO No.	Course Outcome (CO)	Bloom's Category Level
CO 1	Discuss the basic concepts in computer algorithms and their analysis & design using Divide and Conquer.	Level 2: Understand
CO 2	Explain the concepts of Greedy Strategy and Dynamic Programming to use it in solving real world problems.	Level 3: Apply
CO 3	Explain the Branch & Bound technique, Backtracking technique and Lower bounds.	Level 2: Understand
CO 4	Describe the fundamental concepts of Computational Complexity and Network Flows.	Level 2: Understand
CO 5	Discuss the concepts of Approximation and Randomised Algorithms.	Level 2: Understand

Mapping of Course Outcomes with Program Outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	3	3	1	2			2					
CO 2	3	3	1	2			2					
CO 3	3	3	1	2			2					
CO 4	3	3	1	2			2					
CO 5	3	3	1	2			2					

3/2/1: High/Medium/Low

Assessment Pattern

Bloom's Category Levels	Continuous Assessment Tests		End Semester Examination
	1	2	
Level 1: Remember	20	20	20
Level 2: Understand	20	30	30
Level 3: Apply	10		10
Level 4: Analyse			
Level 5: Evaluate			
Level 6: Create			

Mark distribution

Total Marks	Continuous Internal Evaluation (CIE)	End Semester Examination (ESE)	ESE Duration
100	40	60	3 hours

Continuous Internal Evaluation Pattern:

Attendance	: 8 marks
Continuous Assessment Test (2 numbers)	: 20 marks
Assignment/Quiz/Course project	: 12 marks

End Semester Examination Pattern: There will be *two* parts; **Part A** and **Part B**. Part A contain 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer *all* questions. Part B contains 2 questions from each module of which student should answer *any one*. Each question can have a maximum 2 subdivisions and carry 6 marks.

Sample Course Level Assessment Questions

Course Outcome 1 (CO 1):

1. Define "Time Complexity" of an algorithm?
2. What is the need for analysing an algorithm?
3. Define Big Oh Notation.
4. Define the terms Best Case, Worst Case and Average case complexities.
5. Explain the Merge Sort algorithm with an example.

Course Outcome 2 (CO 2):

1. Explain the Greedy Control abstraction.
2. Write the Prim's algorithm and illustrate with an example.
3. State and illustrate the Principle of Optimal Substructure.
4. Explain a solution to the Travelling Salesman problem using Dynamic Programming.

Course Outcome 3 (CO 3):

1. Explain the N-Queen's problem and its solution using Backtracking.
2. Explain the 8-puzzle problem and illustrate how it can be solved using Branch and Bound.
3. Bring out the notion of Decision Trees.
4. What is the lower bound of the time complexity of Comparison based sorting algorithms?

Course Outcome 4 (CO 4):

1. Define class P and NP.
2. What is Polynomial Time Reduction?
3. Show that the Clique problem is NP-Complete.
4. Define the Terms - Flow Network and Network Flow.
5. Explain the Ford-Fulkerson Algorithm.

Course Outcome 5 (CO 5):

1. What is an Approximation algorithm?
2. Describe the 2-approximation algorithm for Vertex Cover problem.
3. What is a Randomised algorithm?
4. Explain the Schwartz-Zippel Lemma. How this Lemma can be used to test the identity of two polynomials.

Model Question Paper

Course Code: 20MCA203

Course Name: Design and Analysis of Algorithms

Max. Marks :60 Duration: 3 Hrs

Part A

Answer all questions. Each question carries 3 marks (10 * 3 = 30 Marks)

1. Define Big Oh notation.
2. Write the control abstraction for a typical Divide and Conquer algorithm.
3. Explain a Greedy strategy which can give the optimal solution for the Knapsack problem.
4. Write a dynamic programming algorithm to compute the factorial of a number.
5. How does Backtracking differ from Branch and Bound?
6. Using a decision tree, show that any search algorithm which searches a given key within an array of n elements must perform at least O(ln n) comparisons in the worst case.
7. What do you mean by the term Polynomial time reduction?
8. Define the term Network Flow and illustrate with an example.

9. What do you mean by approximation ratio of an Approximation algorithm?

10. What is meant by a Randomised Algorithm?

Part B

Answer all questions. Each question carries 6 marks. (5 * 6 = 30 Marks)

11 Write the Linear Search Algorithm and analyse the best, worst and average case complexities of the algorithm. 6

OR

12 Explain the Merge Sort algorithm and give its worst-case analysis. 6

13 Write Kruskal's algorithm to compute the minimum cost spanning tree. 6

OR

14 Explain the dynamic programming algorithm for the Travelling Salesman problem. 6

15 Write the Backtracking algorithm for N-Queen Problem. 6

OR

16 Explain the 8-puzzle problem and its solution using branch and bound technique. 6

17 Show that the Clique problem is NP-Complete. 6

OR

18 Describe the Ford Fulkerson’s procedure to compute the Max-Flow within a given Flow Network. 6

19 Explain the 2-approximation algorithm for Vertex Cover and justify its approximation ratio. 6

OR

20 Describe Randomised Quick sort. 6

Syllabus

Module 1: (8 Hours)

Review of Algorithm Analysis: Time and Space Complexity, Asymptotic Notations, Recurrence Equations, Solving Recurrence Equations- Substitution method and Iteration method.

Divide and Conquer: Control Abstraction, Merge Sort, Quick Sort, Matrix Multiplication.

Module 2: (9 Hours)

Greedy Strategy: Control Abstraction, Knapsack Problem, Minimal Spanning Tree Algorithms- Prim’s and Kruskal’s Algorithm, Job Scheduling with deadlines

Dynamic Programming: Control Abstraction, Principle of Optimal Substructure, All Pairs shortest path problem, Travelling Salesman Problem, Bellman-Ford Algorithm

Module 3:(7 Hours)

Backtracking: Control Abstraction, N-Queens problem, Sum of Subsets Problem

Branch and Bound: Control Abstraction, 8- Puzzle problem

Lower Bounds: The Decision Tree method, Lower Bounds for Comparison based Sort and Searching *(Analysis not required)*

Module 4: (11 Hours)

Complexity Theory: Class P and NP, Polynomial time reductions, Class NP Hard and NP-Complete, Example Problems- Vertex Cover problem, Clique Problem.

Network Flows: Flow Networks and Network Flow, Max- Flow Min Cut Theorem, Ford Fulkerson method, Bipartite matching (*Analysis not required)*

Module 5: (10 Hours)

Introduction to Approximation Algorithms: Approximation Ratio, 2-approximation algorithm for Vertex Cover problem, Vertex Cover Approximation using Linear Programming and LP Rounding Algorithm.

Introduction to Randomised Algorithms: Review of Basic Probability, Schwartz-Zippel Lemma and Polynomial Identity Testing, Randomized Quick Sort (*Proof of Expected Worst Case Analysis not required)*

Text Books

1. Thomas H. Cormen, et al., “Introduction to Algorithms”, Prentice Hall, 3rd Edition (2010)
2. Ellis Horowitz, Sartaj Sahni, Sanguthevar Rajasekaran, “Fundamentals of Computer Algorithms”, Orient Longman, Universities Press, 2nd Edition (2008)

Reference Books

1. Richard Neapolitan, Kumarss Naimipour, “Foundations of Algorithms”, Jones and Bartlett Publishers, Inc, 4th Edition (2011).
2. Sara Baase, Allen Van Gelder, “Computer Algorithms: Introduction to Design and Analysis”, Pearson India, 3rd Edition (2002).
3. A. Levitin, “Introduction to the Design & Analysis of Algorithms”, Pearson Education, 3rd Edition (2008).

Course Contents and Lecture Schedule

Sl. No.	Topic	No. of Lectures
1	**Review of Algorithm Analysis and Divide & Conquer**	**8 Hours**
1.1	Time and Space Complexity	1
1.2	Asymptotic Notations	1
1.3	Recurrence Equations, Solving Recurrence Equations- Substitution method	1
1.4	Iteration method	1
1.5	Divide and Conquer: Control Abstraction, Merge Sort, Merge Sort Analysis	2
1.6	Quick Sort, Quicksort analysis	1
1.7	Matrix Multiplication	1
2	**Greedy Strategy and Dynamic Programming**	**9 Hours**
2.1	Greedy Strategy: Control Abstraction, Knapsack Problem	1
2.2	Minimum Cost Spanning Tree	1
2.3	Prim’s algorithm	1
2.4	Kruskal’s algorithm	1
2.5	Job Scheduling with deadlines	1
2.6	Dynamic Programming: Control Abstraction, Principle of Optimal substructure	1
2.7	All Pairs shortest path problem	1
2.8	Travelling Salesman Problem	1
2.9	Bellman-Ford Algorithm	1

3	**Backtracking, Branch & Bound, Lower Bounds**	**7 Hours**
3.1	Backtracking: Control Abstraction N- Queens problem	1
3.2	Sum of subsets problem	1
3.3	Branch and Bound: Control Abstraction 8- Puzzle problem	1
3.4	Lower Bounds: The Decision Tree method	2
3.5	Lower Bounds for Comparison based Sorting	1
3.6	Lower bounds for searching	1
4	**Computational complexity, Network Flows**	**11 Hours**
4.1	Class P, NP	1
4.2	Polynomial Time Reductions	1
4.3	Class NP-Hard and NP-Complete	2
4.4	Vertex Cover Problem	1
4.5	Clique problem	1
4.6	Flow Networks and Network Flows	2
4.7	Max Flow Min Cut Theorem	1
4.8	Ford Fulkerson’s method	1
4.9	Bipartite matching	1
5	**Approximation & Randomised Algorithms**	**10 Hours**
5.1	Approximation algorithms- introduction, Approximation Ratio	1
5.2	2- approximation algorithm for Vertex Cover problem	1
5.3	Vertex Cover Approximation using Linear Programming and LP Rounding Algorithm	2
5.4	Randomized Algorithms: introduction, Review of Basic Probability	1
5.5	Review of Basic probability	2
5.6	Schwartz-Zippel Lemma and Polynomial Identity Testing	2
5.7	Randomized Quick Sort	1

CODE	COURSE NAME	CATEGORY	L	T	P	CREDIT
20MCA261	OPERATIONS RESEARCH	ELECTIVE	3	1	0	4

Preamble: This course introduces the concepts of linear programming problems. The topics treated in this course have applications in real life problems.

Prerequisite: Nil

Course Outcomes: After completion of the course the student will be able to

CO No.	Course Outcome (CO)	Bloom's Category Level
CO 1	Solve different types of Linear Programming Problems.	Level 3: Apply
CO 2	Apply the concept of linear programming problems in real life.	Level 3: Apply
CO 3	Solve different decision-making problems using optimization techniques.	Level 3: Apply
CO 4	Use PERT and CPM to analyse project network management.	Level 3: Apply
CO 5	Identify suitable queuing model and solve queuing problems.	Level 3: Apply

Mapping of Course Outcomes with Program Outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	3	3	1	-	-	-	2	-	-	-	-	-
CO 2	3	3	3	-	-	-	2	-	-	-	-	-
CO 3	3	3	3	-	-	-	2	-	-	-	-	-
CO 4	3	3	1	1	-	-	2	2	-	-	-	-
CO 5	3	3	3	-	-	-	2	-	-	-	-	-

3/2/1: High/Medium/Low

Assessment Pattern

Bloom's Category Levels	Continuous Assessment Tests		End Semester Examination
	1	**2**	
Level 1: Remember	10	10	10
Level 2: Understand	20	20	20
Level 3: Apply	20	20	30
Level 4: Analyse			
Level 5: Evaluate			
Level 6: Create			

Mark Distribution

Total Marks	Continuous Internal Evaluation (CIE)	End Semester Examination (ESE)	ESE Duration
100	40	60	3 hours

Continuous Internal Evaluation Pattern:

Attendance	: 8 marks
Continuous Assessment Test (2 numbers)	: 20 marks
Assignment/Quiz/Course project	: 12 marks

End Semester Examination Pattern: There will be two parts; Part A and Part B. Part A contain 10 compulsory short answer questions, 2 from each module. Each question carries 3 marks. Part B contains 2 questions from each module of which student should answer any one. Each question can have maximum 2 sub-divisions and carry 6 marks

Sample Course Level Assessment Questions

Course Outcome 1 (CO 1):

1. Define slack variable, surplus variable and optimal basic feasible solution.
2. Obtain all basic feasible solution of the set of equations:

   a) $2x_1 + 3x_2 + 4x_3 + x_4 = 2$

   b) $x_1 + x_2 + 7x_3 + x_4 = 4$
3. Solve by Big M method

   Maximise $Z = 6x_1 - 3x_2 + 2x_3$

   Subject to $2x_1 + x_2 + x_3 \leq 16$

   $3x_1 + 2x_2 + x_3 \leq 18$

$$x_1 - 2x_2 \geq 8$$

$$x_1,\ x_2,\ x_3 \geq 0$$

Course Outcome 2 (CO 2):

1. Construct the dual of

$$\begin{array}{ll} \text{Maximise} & Z = 3x_1 + 17x_2 + 9x_3 \\ \text{Subject to} & x_1 - x_2 + x_3 \geq 3 \\ & -3x_1 + 2x_2 \leq 1 \end{array}$$

$$x_1,\ x_2,\ x_3 \geq 0$$

2. Prove that the dual of the dual is the primal
3. Solve using the principle of duality

$$\begin{array}{ll} \text{Minimise} & Z = 3x_1 + 5x_2 \\ \text{Subject to} & 2x_1 + 8x_2 \geq 40 \\ & 3x_1 + 4x_2 \geq 50 \end{array}$$

$$x_1,\ x_2 \geq 0$$

Course Outcome 3 (CO 3):

1. Explain North West Corner method
2. Solve the following transportation problem

	1	2	3	Supply
1	2	7	4	5
2	3	3	1	8
3	5	4	7	7
4	1	6	2	14
Demand	7	9	18	34

3. Solve the assignment problem

	I	II	III	IV
A	16	10	14	11
B	14	11	15	15
C	15	15	13	12
D	13	12	14	15

Course Outcome 4 (CO 4):

1. Explain critical path analysis.
2. A project consists of series of tasks labelled A, B, …, H, I with the following relationships (W < X, Y means X and Y cannot start until W is completed; X, Y < W means W cannot start until both X and Y are completed). With this notation construct the network diagram having the following constraints:

   A < D, E; B, D < F; C < G; B, G< H; F, G < I.

   Find also the minimum time of completion of the project, when the time (in days) of completion of each task is as follows:

Task :	A	B	C	D	E	F	G	H	I
Time :	23	8	20	16	24	18	19	4	10

3. A project consists of eight activities with the following relevant information.

Activity	Immediate predecessor	Estimated duration (days)		
		Optimistic	Most likely	Pessimistic
A	--	1	1	7
B	--	1	4	7
C	--	2	2	8
D	A	1	1	1
E	B	2	5	14
F	C	2	5	8
G	D, E	3	6	15
H	F, G	1	2	3

(i) Draw the PERT network and find out the expected project completion time.

(ii) What duration will have 95% confidence for project completion?

(iii) If the average duration for activity F increases to 14 days, what will be its effects on the expected project completion time which will have 95% confidence?

(For standard normal Z = 1.645, area under the standard normal curve from 0 to Z is 0.45)

Course Outcome 5 (CO 5):

1. Explain Birth-death process.
2. In a railway marshalling yard, goods trains arrive at a rate of 30 trains per day. Assuming that the inter-arrival time follows an exponential distribution and the

service time distribution is also exponential with an average 36 minutes. Calculate the following:

i. The mean queue size (line length), and

ii. The probability that the queue size exceeds 10.

iii. If the input of trains increases to an average 33 per day, what will be the change in (i) and (ii)?

3. At a railway station, only one train is handled at a time. The railway yard is sufficient only for two trains to wait while other is given signal to leave the station. Trains arrive at the station at an average rate of 6 per hour and the railway station can handle them on an average of 12 per hour. Assuming Poisson arrivals and exponential service distribution, find the steady-state probabilities for the various number of trains in the system. also find the average waiting time of a new train coming into the yard

Model Question Paper
Course Code: 20MCA261
Course name: Operations Research

Max. Marks: 60 Duration: 3hrs

Part A

Answer all questions, each carries 3 marks (10×3 = 30)

1. Write down the basic structure of a linear programming problem in the mathematical form.
2. Define slack and surplus variables in LPP.
3. State the fundamental theorem of duality.
4. Write the dual of the following

$$\begin{aligned} &Max\ Z = x_1 - x_2 + 3x_3 \\ &subject\ to\ x_1 + x_2 + x_3 \le 10 \\ &\qquad 2x_1 - x_3 \le 2 \\ &\qquad 2x_1 - 2x_2 + 3x_3 \le 6 \\ &\qquad x_1, x_2, x_3 \ge 0 \end{aligned}$$

5. Obtain the IBFS using north west corner method

	D_1	D_2	D_3	D_4	Supply
O_1	2	4	3	6	20
O_2	7	3	8	2	10
O_3	2	2	9	11	15
Demand	15	15	8	7	

6. Describe the Matrix Minima method.
7. What is queue discipline?
8. Explain single serve Poisson queuing model with infinite capacity.
9. Activities P, Q and R instantly follow activity M, and their current starting times are 12, 19 and 10. So, what is the latest finishing time for activity M?
10. What is the difference between PERT and CPM.

Part B

Answer all questions, each carries 6 marks (5×6 = 30)

11. Solve the following problem by Simplex method

$$Max\ Z = 5x_1 + 3x_2$$
$$subject\ to\ 4x_1 - x_2 \leq 10$$
$$2x_1 + 2x_2 \leq 50$$
$$x_1, x_2 \geq 0$$

or

12. Solve by Big-M method

$$Max\ Z = 6x_1 - 3x_2 + 2x_3$$
$$subject\ to\ 2x_1 + x_2 + x_3 \leq 16$$
$$3x_1 + 2x_2 + x_3 \leq 18$$
$$x_2 - 2x_3 \geq 8$$
$$x_1, x_2, x_3 \geq 0$$

13. Prove that the dual of a dual is the primal.

or

14. Solve the following by using the dual principle

$$Max\ Z = 40x_1 + 35x_2$$
$$subject\ to\ 2x_1 + 3x_2 \leq 60$$
$$4x_1 + 3x_2 \leq 96$$
$$x_1, x_2 \geq 0$$

15. Solve the following Assignment problem

	I	II	III	IV
A	2	3	4	5
B	4	5	6	7
C	7	8	9	8
D	3	5	8	9

or

16. Solve the following transportation problem

	D_1	D_2	D_3	D_4	Supply
O_1	5	2	4	3	22
O_2	4	5	1	6	15
O_3	4	6	7	5	8
Demand	7	12	17	9	

17. Explain critical path analysis.

or

18. A project consists of eight activities with the following relevant information.

Activity	Immediate predecessor	Estimated duration (days)		
		Optimistic	Most likely	Pessimistic
A	--	1	1	7
B	--	1	4	7
C	--	2	2	8
D	A	1	1	1
E	B	2	5	14
F	C	2	5	8
G	D, E	3	6	15
H	F, G	1	2	3

(iv) Draw the PERT network and find out the expected project completion time.

(v) What duration will have 95% confidence for project completion?

(vi) If the average duration for activity F increases to 14 days, what will be its effects on the expected project completion time which will have 95% confidence?

(For standard normal Z = 1.645, area under the standard normal curve from 0 to Z is 0.45)

19. Explain birth-death process.

or

20. At a railway station, only one train is handled at a time. The railway yard is sufficient only for two trains to wait while other is given signal to leave the station. Trains arrive at the station at an average rate of 6 per hour and the railway station can handle them on an average of 12 per hour. Assuming Poisson arrivals and exponential service distribution, find the steady-state probabilities for the various number of trains in the system. also find the average waiting time of a new train coming into the yard.

Syllabus

Module 1: (9 Hours)

Linear programming problem- Slack and surplus variable- Standard form- Solution of Linear programming problem- Basic solution- Basic feasible solution- Degenerate- and Non-degenerate solutions- Optimal solution- Solution by simplex method- Artificial variables- Big- M method.

Module 2: (9 Hours)

Duality in Linear Programming Problem- Statement of duality theorem- Statement of complementary slackness theorem. The primal- Duality solutions using simplex method- Revised simplex method

Module 3: (9 Hours)

Transportation problem- Solution of Transportation problem- Finding an initial basic feasible solution- North West Corner method- Matrix minima method- Vogel's Approximation method- Test for Optimality- Modi method- Unbalanced Transportation problem- Maximisation in Transportation problem. Assignment problem- Optimal solution- Hungarian method of assignment- Maximization in assignment problem.

Module 4: (9 Hours)

Network analysis- Project scheduling- Construction of project networks- Critical path method (CPM)- Identification of critical path using CPM- Estimation of Floats- Total float- Independent float- Project Evaluation and Review Technique (PERT) - Computation of expected completion times by PERT.

Module 5: (9 Hours)

Queuing theory- Elements of Queuing System- Kendall's notation- Operating characteristics- Poisson process- Exponential distribution- Mean and variance- Birth and Death process. Queuing models based on Poisson process- Single server models with finite and infinite capacity- Multi server model with finite and infinite capacity.

Note:

- Programming Assignments using Python and appropriate Case Studies may be given at the end of each module.

- Linear Programming Problems in module 1 and module 2 and Transportation problems in module 3 can be solved using Python library PuLP. Using Numpy, PERT/CPM problems in module 4 can be solved.

Text Book

1. KantiSwarup, P.K. Gupta and Man Mohan, Operation Research, Sultan Chand (2010)

Reference Books

1. Hamdy A Taha, Operations Research- an introduction, Eighth Edition, Prentice Hall of India.
2. Ravindran, Philips and Solberg, Wiley, Operation Research, Second edition (2007)

Web References

1. https://pypi.org/project/PuLP/
2. https://numpy.org/

Course Contents and Lecture Schedule

Sl. No.	Topic	No. of Lectures
1	**Module 1**	**9 Hours**
1.1	Linear programming problem- Slack and surplus variable- Standard form	1
1.2	Solution of Linear programming problem- Basic solution- Basic feasible solution- Degenerate- and Non-degenerate solutions- Optimal solution	2
1.3	Solution by simplex method	3
1.4	Artificial variables- Big- M method	3
2	**Module 2**	**9 Hours**
2.1	Duality in Linear Programming Problem	1
2.2	Statement of duality theorem- Statement of complementary slackness theorem	2
2.3	The primal- Duality solutions using simplex method	3
2.4	Revised simplex method	3
3	**Module 3**	**9 Hours**
3.1	Transportation problem- Solution of Transportation problem- Finding an initial basic feasible solution- North West Corner method	2
3.2	Matrix minima method- Vogel’s Approximation method	1
3.3	Test for Optimality- Modi method- Unbalanced Transportation problem- Maximisation in Transportation problem	3
3.4	Assignment problem- Optimal solution- Hungarian method of assignment- Maximization in assignment problem	3
4	**Module 4**	**9 Hours**
4.1	Network analysis- Project scheduling- Construction of project networks	1
4.2	Critical path method (CPM)- Identification of critical path using CPM	2
4.3	Estimation of Floats- Total float- Independent float	3
4.4	Project Evaluation and Review Technique (PERT)	2
4.5	Computation of expected completion times by PERT	1

5	**Module 5**	**9 Hours**
5.1	Queuing theory- Elements of Queuing System- Kendall’s notation- Operating characteristics- Poisson process	1
5.2	Exponential distribution- Mean and variance- Birth and Death process	2
5.3	Queuing models based on Poisson process	3
5.4	Single server models with finite and infinite capacity	1
5.5	Multi server model with finite and infinite capacity	2

CODE	COURSE NAME	CATEGORY	L	T	P	CREDIT
20MCA263	**CYBER SECURITY & CRYPTOGRAPHY**	**ELECTIVE**	**3**	**1**	**0**	**4**

Preamble: This course is designed to provide theoretical concepts used in cryptography and to introduce the students to various cryptographic algorithms and techniques used for implementing data security and protection. This course also discusses common web application security vulnerabilities.

Prerequisite: Student is expected to have studied mathematics courses that cover Elementary Number Theory, Finite Field, Discrete Logarithm and Euclidean Algorithm.

Course Outcomes: After completion of the course the student will be able to

CO No.	Course Outcome (CO)	Bloom's Category Level
CO 1	Explain various types of security attacks, security mechanisms, security services and classical encryption techniques.	Level 2: Understand
CO 2	Make use of Symmetric and Asymmetric encryption techniques to solve cryptographic problems.	Level 3: Apply
CO 3	Describe the concepts of message authentication codes, hash functions and digital signing techniques for ensuring secure transactions.	Level 2: Understand
CO 4	Discuss security services in Application, Transport and Network layers.	Level 2: Understand
CO 5	Explain common web application security vulnerabilities and various prevention mechanisms.	Level 2: Understand

Mapping of Course Outcomes with Program Outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	2	1	1				1					
CO 2	2	2	2	1			1					
CO 3	2	1	1				1					
CO 4	2	1	1			2	1					
CO 5	2	1	1			2	1					

3/2/1: High/Medium/Low

Assessment Pattern

Bloom's Category Levels	Continuous Assessment Tests		End Semester Examination
	1	2	
Level 1: Remember	15	15	20
Level 2: Understand	35	35	40
Level 3: Apply			
Level 4: Analyse			
Level 5: Evaluate			
Level 6: Create			

Mark distribution

Total Marks	Continuous Internal Evaluation (CIE)	End Semester Examination (ESE)	ESE Duration
100	40	60	3 hours

Continuous Internal Evaluation Pattern:

Attendance	: 8 marks
Continuous Assessment Test (2 numbers)	: 20 marks
Assignment/Quiz/Course project	: 12 marks

End Semester Examination Pattern: There will be two parts; Part A and Part B. Part A contains 10 compulsory short answer questions, 2 from each module. Each question carries 3 marks. Part B contains 2 questions from each module of which student should answer any one. Each question can have a maximum of 2 sub-divisions and carry 6 marks.

Sample Course Level Assessment Questions

Course Outcome 1 (CO 1):

1. Briefly explain each component of OSI security architecture.
2. Compare Substitution and Transposition techniques in cryptography.
3. Explain how steganography is used in cryptography.

Course Outcome 2 (CO 2):

1. Explain block cipher modes of operation.
2. Compare DES and AES

3. Perform encryption and decryption using RSA Algorithm with parameters: P=17, q = 11, e = 7, M = 88.

Course Outcome 3 (CO 3):

1. Compare the features of HMAC and CMAC algorithms.
2. Explain important steps in DSS.
3. Describe the terms (a) birthday attack (b) hashcash (c) blind signature

Course Outcome 4 (CO 4):

1. Explain any one protocol used in E-mail for security.
2. Explain how security is provided in Network Layer using IPsec.
3. Describe the process of securing electronic transactions.

Course Outcome 5 (CO 5):

1. Discuss any four Application Security Risks.
2. Which are the different forms of XSS and how to prevent these?
3. Explain the attack scenario of any four web application security vulnerabilities.

Model Question Paper

Course Code: 20MCA263

Course Name: CYBER SECURITY & CRYPTOGRAPHY

Max. Marks :60 **Duration: 3 Hrs**

Part A

Answer all questions.

Each question carries 3 marks (10 x 3 = 30 Marks)

.

1. Compare phishing and ransomware attacks.
2. What is OSI security architecture?
3. List out the advantages and disadvantages of Output Feed Back mode.
4. Explain round functions used in DES.
5. Explain important steps in DSS.
6. Describe the terms (a) birthday attack (b) hashcash (c) blind signature.
7. Describe security association of IPSec.
8. Explain about S/MIME.

9. How can we prevent Injection attack?

10. What is XXE? How to prevent it?

(10 x 3=30 marks)

Part B

Answer all questions. Each question carries 6 marks. (5 * 6 = 30 Marks)

11. Explain Network security model with the help of a neat diagram (6)

OR

12. Describe the working of Playfair cipher and Hill cipher. (6)

13. Apply Diffie-Hellman key exchange algorithm to compute the shared private key using the values P = 23, g = 9, a= 4, b= 3. Explain the steps in detail. (6)

OR

14. Perform encryption and decryption using RSA Algorithm with parameters: P=17, q = 11, e = 7, M = 88. Explain the steps in detail. (6)

15. Compare HMAC and CMAC protocol with suitable diagrams. (6)

OR

16. Compare various signature schemes with suitable diagrams. (6)

17. Explain PGP cryptographic functions with diagram. (6)

OR

18. Explain Secure Electronic Transaction Protocol. (6)

19. Briefly explain any four Application Security Risks. (6)

OR

20. Explain the attack scenarios of any four web application security vulnerabilities. (6)

(5 x 6=30 Marks)

Syllabus

Module 1: (7 Hours)
Introduction to Cryptography, OSI security architecture: Security Services, Mechanisms and attacks- Phishing, Ransomware, DoS attack. Network security model. Classical Encryption techniques - Symmetric cipher model, substitution techniques, transposition techniques. Steganography.
Module 2: (10 Hours)
Conventional Symmetric Key Encryption: Block ciphers and Stream Ciphers, Block Cipher Design Principles, Modes of operation, Data Encryption Standard, Advanced Encryption Standard (AES), Multiple Encryption, Triple DES. **Public key cryptography:** Principles of public key cryptosystems-The RSA algorithm-Key management – Diffie Hellman Key exchange - Elliptic curve arithmetic - Elliptic curve cryptography.
Module 3: (10 Hours)
Hash Functions and MAC: Properties of hash functions, birthday attack, hashcash, Message Authentication Code Algorithms, MAC protocols: HMAC, CMAC. **Digital Signatures:** Classification of signature schemes: RSA signature, Digital Signature Standard, Overview of ElGamal and Schnorr schemes, One time signature schemes, Attacks on Digital Signatures, Blind Signatures.
Module 4: (10 Hours)
Introduction to Cyber Security: Email Security: Security Services for email, Attacks possible through email, Establishing keys privacy, authentication of the source, Message Integrity, Non-repudiation, Pretty Good Privacy, S/MIME. **IP Security:** Overview of IPSec, IPv4 and IPv6, Authentication Header, Encapsulation Security Payload (ESP), Internet Key Exchange. **Transport Level Security:** SSL/TLS Basic Protocol, computing the keys, client authentication, PKI as deployed by SSL, Attacks fixed in v3, Exportability, Encoding, Secure Electronic Transaction (SET).
Module 5: (8 Hours)
Common web application security vulnerabilities: Injection flaws, Broken authentication, Sensitive data exposure, XML External Entities (XXE), Broken access control, Security misconfiguration, Cross-Site Scripting (XSS), Insecure deserialization, Using components with known vulnerabilities, Insufficient logging & monitoring. Example attack scenarios of each of the vulnerabilities listed; how to prevent them

Text Book

1. William Stallings, “Cryptography and Network Security,” 6th Edition, Pearson Education, March (2013).
2. Behrouz A. Forouzan, “Introduction to Cryptography and Network Security”, Tata McGraw-Hill Publishing 2nd Edition (2011).

Reference Books

1. Charlie Kaufman, Radia Perlman and Mike Speciner, “Network Security”, Prentice Hall of India, 2002.
2. Manuel Mogollon, “Cryptography and Security Services – Mechanisms and Applications”, Cybertech Publishing, 2008
3. William R. Cheswick, Steven M. Bellovin, Aviel D. Rubin, “Firewalls and Internet Security” Addison- Wesley, 2003

Web References

1. http://www.hashcash.org/hashcash.pdf [Reference for hashcash]
2. https://owasp.org/www-pdf-archive/OWASP_Top_10-2017_%28en%29.pdf. [Reference for Module 5]
3. https://www.coursera.org/learn/crypto
4. https://www.coursera.org/learn/crypto2

Course Contents and Lecture Schedule

Sl. No.	Topic	No. of Lectures
1	**Introduction to Cryptography**	**7 Hours**
1.1	What is cryptography, Related Terms, Need of cryptosystems	1
1.2	OSI security architecture: Security Services, Mechanisms	1
1.3	Security attacks- Phishing, Ransomware, DoS attack.	1
1.4	Network security model	1
1.5	Classical Encryption techniques, Symmetric cipher model	1
1.6	Substitution techniques	1
1.7	Transposition techniques, Steganography	1
2	**Conventional Symmetric and Public Key Encryption**	**10 Hours**
2.1	Block ciphers and Stream Ciphers, Block Cipher Design Principles	1
2.2	Modes of operation	1
2.3	Data Encryption Standard	1
2.4	Advanced Encryption Standard (AES)	1
2.5	Multiple Encryption, Triple DES	1
2.6	Public key cryptography: Principles of public key cryptosystems	1
2.7	The RSA algorithm	1

2.8	Key management	1
2.9	Diffie Hellman Key exchange	1
2.10	Elliptic curve arithmetic - Elliptic curve cryptography.	1
3	**Hash Functions and MAC**	**10 Hours**
3.1	Properties of hash functions, birthday attack	1
3.2	Hashcash, Message Authentication Code Algorithms	1
3.3	MAC protocols: HMAC, CMAC	1
3.4	Digital Signatures: Classification of signature schemes	1
3.5	RSA signature	1
3.6	Digital Signature Standard	1
3.7	Overview of ElGamal and Schnorr schemes	1
3.8	One time signature schemes	1
3.9	Attacks on Digital Signatures	1
3.10	Blind Signatures	1
4	**Introduction to Cyber Security**	**10 Hours**
4.1	Email Security: Security Services for email, Attacks possible through email	1
4.2	Establishing keys privacy, authentication of the source, Message Integrity, Non-repudiation	1
4.3	Pretty Good Privacy, S/MIME	1
4.4	IP Security: Overview of IPSec	1
4.5	IPv4 and IPv6, Authentication Header	1
4.6	Encapsulation Security Payload (ESP), Internet Key Exchange	1
4.7	Transport Level Security: SSL/TLS Basic Protocol	1
4.8	computing the keys, client authentication, PKI as deployed by SSL	1
4.9	Attacks fixed in v3, Exportability, Encoding	1
4.10	Secure Electronic Transaction (SET)	1
5	**Common web application security vulnerabilities**	**8 Hours**
5.1	Common web application security vulnerabilities	1
5.2	Injection flaws, Broken authentication	1
5.3	Sensitive data exposure, XML External Entities (XXE)	1
5.4	Broken access control, Security misconfiguration	1
5.5	Cross-Site Scripting (XSS), Insecure deserialization	1
5.6	Using components with known vulnerabilities, Insufficient logging & monitoring.	1
5.7	Example attack scenarios of each of the vulnerabilities listed	1
5.8	How to prevent each of the vulnerabilities.	1

CODE	COURSE NAME	CATEGORY	L	T	P	CREDIT
20MCA265	Cloud Computing	ELECTIVE	3	1	0	4

Preamble: The syllabus is prepared with a view to equip the students to learn basic concepts in cloud computing - compute, storage, networking. They should gain basic understanding of orchestration, HA and failover.

Prerequisite: Awareness in Virtualisation and Containers is desirable.

Course Outcomes: After completion of the course the student will be able to

CO No.	Course Outcome (CO)	Bloom's Category Level
CO 1	Understand the basic concepts in cloud computing and OpenStack logical architecture	Level 2: Understand
CO 2	Discuss OpenStack cloud controller and common services	Level 3: Apply
CO 3	Compare different OpenStack compute service components and storage types	Level 2: Understand
CO 4	Describe the OpenStack Networking- Connection types and networking services	Level 2: Understand
CO 5	Discuss orchestration, HA and failover in OpenStack	Level 2: Understand

Mapping of Course Outcomes with Program Outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	2				3		2					1
CO 2	2				3		2					1
CO 3	2				3		2					1
CO 4	2				3		2					1
CO 5	2				3		2					1

3/2/1: High/Medium/Low

Assessment Pattern

Bloom's Category Levels	Continuous Assessment Tests		End Semester Examination
	1	2	
Level 1: Remember	20	15	20
Level 2: Understand	20	35	30
Level 3: Apply	10		10
Level 4: Analyse			
Level 5: Evaluate			
Level 6: Create			

Mark distribution

Total Marks	Continuous Internal Evaluation (CIE)	End Semester Examination (ESE)	ESE Duration
100	40	60	3 hours

Continuous Internal Evaluation Pattern:

Attendance	: 8 marks
Continuous Assessment Test (2 numbers)	: 20 marks
Assignment/Quiz/Course project	: 12 marks

End Semester Examination Pattern: There will be two parts; Part A and Part B. Part A contains 10 compulsory short answer questions, 2 from each module. Each question carries 3 marks. Part B contains 2 questions from each module of which student should answer any one. Each question can have a maximum 2 subdivisions and carry 6 marks.

Sample Course Level Assessment Questions

Course Outcome 1 (CO 1):

1. List and explain various components of Nova compute service.
2. Explain the neutron architecture?
3. Briefly describe keystone identity management.

Course Outcome 2 (CO 2):

1. Explain the telemetry services in OpenStack.
2. Explain the steps involved in bringing up a working OpenStack Ansible on the deployment host.
3. Explain the steps in network configuration

Course Outcome 3 (CO 3):

1. Explain briefly swift architecture
2. Briefly explain how data is handled in the cluster by swift
3. What is meant by CPU over commitment?

Course Outcome 4 (CO 4):

1. Explain steps in associating a floating IP to a virtual machine.
2. Briefly explain the steps in creating a virtual network with two subnets
3. Briefly explain Linux bridge-based connectivity?

Course Outcome 5 (CO 5):

1. Briefly explain the major components in heat?
2. Explain the different metrics that can be measured in a highly available infrastructure?
3. Explain the need for Service level agreement.

Model Question Paper

Course Code: 20MCA265

Course Name: Cloud Computing

Max. Marks :60 **Duration: 3 Hrs**

Part A

Answer all questions. Each question carries 3 marks (10 * 3 = 30 Marks)

1. What are the different components in OpenStack logical architecture?
2. Differentiate between private cloud and public cloud.
3. Explain asymmetric clustering and symmetric clustering.
4. List out the functionalities handled by the cloud controller.
5. Briefly explain docker containers.
6. Compare object storage with NAS/SAN based storage.
7. Describe the steps in connecting two networks using a virtual router.
8. Write a short note on firewall as a service
9. List the HA levels in OpenStack.
10. Explain the purpose of HA proxy.

Part B

Answer all questions. Each question carries 6 marks. (5 * 6 = 30 Marks)

11 List and explain the different components in OpenStack Architecture. 6

OR

12 a. Explain the provisioning of VM in OpenStack using a diagram 4

b. Describe the best practices used in Physical mode design 2

13 Explain the keystone architecture 6

OR

14 Explain the steps involved in running OpenStack playbooks 6

15 Explain in detail the multiple services involved in launching an instance 6

OR

16 Explain the steps in deploying swift service 6

17 Explain the architecture of neutron in detail 6

OR

18 Explain the categorization of neutron virtual networks in detail 6

19 Explain stacking in OpenStack 6

OR

20 Explain in detail steps involved in setting a database with high availability 6

Syllabus

Module 1: Overview of OpenStack (7 Hours)

Introduction to cloud computing, private cloud, public cloud, hybrid cloud architecture. Cloud Services - Infrastructure as a Service, Platform as a Service, Storage as a Service. Designing OpenStack Cloud Architectural Consideration - OpenStack - The new data centre paradigm - OpenStack logical architecture - Nova - Compute Service-Neutron - Networking services - Gathering the pieces and building a picture - A sample architecture setup.

Module 2: OpenStack cluster - Controller and common services (6 Hours)

OpenStack Cluster – The Cloud Controller and Common Services- Asymmetric clustering, Symmetric clustering, The cloud controller - The keystone service.

The nova-conductor service, The nova-scheduler service, The API services, Image management, The network service, The horizon dashboard, The telemetry services.

Module 3: OpenStack compute and Storage (12 Hours)
OpenStack Compute -The compute service components - Deciding on the hypervisor - OpenStack Magnum Project - Segregating the compute cloud - Overcommitment considerations - Storing instances' alternatives - Understanding instance booting - Planning for service recovery. OpenStack Storage - Block, Object, and File Share - Understanding the storage types - Ephemeral Storage - Persistent storage - A spotlight on Swift - Deploying Swift service - Using block storage service: Cinder.
Module 4: OpenStack Networking (10 Hours)
The architecture of Neutron - Implementing virtual networks - Connecting virtual networks with routers - Implementing network security in OpenStack. OpenStack Networking - The architecture of Neutron - Implementing virtual networks - VLAN, Tunnel based, Virtual Switches, The ML2 Plugin. Neutron Subnets - Connecting virtual networks with routers - Configuring the routing service - connecting networks using a virtual router, connecting to the external world, connectivity from the external world, associating a floating IP - Implementing network security in OpenStack
Module 5: OpenStack Orchestration, HA and failover (10 Hours)
Orchestration in OpenStack - Heat and its Components, stacking in OpenStack, OpenStack Orchestration with Terraform. OpenStack HA and failover: Scope of HA in OpenStack, HA in the database, HA in the Queue, Implementing HA on RabbitMQ.

Text Book

1. Omar Khedher, Chandan Datta Chowdhury, Mastering OpenStack, 2nd Edition, Packt Publishing, 2017

Reference Books

1. Tom Fifield, Diane Fleming, Anne Gentle,Lorin Hochstein, Jonathan Proulx, Everett Toews, and Joe Topjian, OpenStack Operations Guide, O'REILY,1/e, 2014.
2. Uchit Vyas, Applied OpenStack Design Patterns, Apress, 1/e, 2016.
3. V. K. Cody Bumgardner, OpenStack in action, Manning, 2016.
4. Amar Kapadia, Sreedhar Varma, Kris Rajana, Implementing Cloud Storage with OpenStack Swift, Packt Publishing, 2014.
5. https://docs.openstack.org/wallaby/?_ga=2.231002015.1428061357.1620834394-1139122985.1620834394

Course Contents and Lecture Schedule

Sl. No.	Topic	No. of Lectures
1	**Overview of OpenStack**	**7 Hours**
1.1	Introduction to cloud computing, private cloud, public cloud, hybrid cloud architecture.	1
1.2	Cloud Services - Infrastructure as a Service, Platform as a Service, Storage as a Service	1
1.3	Designing OpenStack Cloud Architectural Consideration - OpenStack - The new data center paradigm -OpenStack logical architecture	1
1.4	Nova - Compute service	1
1.5	Neutron - Networking services	1
1.6	Gathering the pieces and building a picture	1
1.7	A sample architecture setup	1
2	**OpenStack cluster - Controller and common services**	**6 Hours**
2.1	OpenStack Cluster – The Cloud Controller and Common Services- Asymmetric clustering, Symmetric clustering	1
2.2	The cloud controller - The keystone service	2
2.3	The nova-conductor service, The nova-scheduler service, The API services, Image management.	1
2.4	The network service, The horizon dashboard, The telemetry services	2
3	**OpenStack compute and Storage**	**12 Hours**
3.1	The compute service components-Deciding on the hypervisor-OpenStack Magnum project	1
3.2	Segregating the compute cloud	1
3.3	Overcommitment considerations	1
3.4	Storing instances' alternatives	1
3.5	Understanding instance booting	1
3.6	Planning for service recovery	1
3.7	OpenStack Storage - Block, Object, and File Share-Understanding the storage types	1
3.8	A spotlight on swift	2
3.9	Deploying swift service	1

3.10	Using Block Storage Service Cinder	2
4	**OpenStack Networking**	**10 Hours**
4.1	The architecture of Neutron	1
4.2	Implementing virtual networks - VLAN, Tunnel based	1
4.3	Virtual Switches, The ML2 Plugin	1
4.4	Neutron Subnets	2
4.5	Connecting virtual networks with routers - Configuring the routing service	1
4.6	Connecting networks using a virtual router, Connecting to the external world	1
4.7	Connectivity from the external world, Associating a floating IP	1
4.8	Implementing network security in OpenStack	2
5	**OpenStack Orchestration, HA and Failover**	**10 Hours**
5.1	Orchestration in OpenStack, Heat and its Components	1
5.2	Stacking in OpenStack	2
5.3	OpenStack Orchestration with Terraform	2
5.4	Scope of HA in OpenStack	2
5.5	HA in the database	1
5.6	HA in the Queue, Implementing HA on RabbitMQ	2

Suggested Assignments

1) Create VMs in your physical machine using OpenStack to set up the following services: Moodle, MySQL Server, Samba. Design the desired configuration of the physical machine to handle the requirements of the entire college.
2) Set up storage services for storing external files for Moodle.
3) Set up firewall rules for samba, MySQL server, allow the connection to MySQL server only to Moodle VM.
4) Set up recovery plans for the above services
5) Convert the MySQL server to HA MySQL server.
6) Setup a load balancer for the Moodle server.

CODE	COURSE NAME	CATEGORY	L	T	P	CREDIT
20MCA267	**CYBER FORENSICS**	**ELECTIVE**	**3**	**1**	**0**	**4**

Preamble: This course helps the learner to understand the fundamentals of cyber forensics. Student will learn common approaches, practices and techniques used for collecting and preserving digital evidences in this course.

Prerequisite: Basic knowledge in operating systems & computer networks.

Course Outcomes: After completion of the course the student will be able to

CO No.	Course Outcome (CO)	Bloom's Category Level
CO 1	Explain a computer crime and the concept of rules or policy violations.	Level 2: Understand
CO 2	Gather evidences and preserve the collected evidence with the required knowledge on various storage format choices.	Level 3: Apply
CO 3	Describe digital storage and file systems and extract data using Autopsy.	Level 3: Apply
CO 4	Explain mobile device forensics and practice data acquisition procedures for network forensics using Wireshark.	Level 3: Apply
CO 5	Prepare forensics reports both using tools and manually and explain ethics and code for expert witness.	Level 2: Understand

Mapping of Course Outcomes with Program Outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	2	1		1		2	1		2			
CO 2	2	1		1	2		1					
CO 3	2	1		1	2		1					
CO 4	2	1		1	2		1					
CO 5	2	1		1	2	3	1		1			

3/2/1: High/Medium/Low

Assessment Pattern

Bloom's Category Levels	Continuous Assessment Tests		End Semester Examination
	1	2	
Level 1: Remember	15	15	20
Level 2: Understand	35	35	40
Level 3: Apply			
Level 4: Analyse			
Level 5: Evaluate			
Level 6: Create			

Mark distribution

Total Marks	Continuous Internal Evaluation (CIE)	End Semester Examination (ESE)	ESE Duration
100	40	60	3 hours

Continuous Internal Evaluation Pattern:

Attendance	: 8 marks
Continuous Assessment Test (2 numbers)	: 20 marks
Assignment/Quiz/Course project	: 12 marks

End Semester Examination Pattern: There will be two parts; Part A and Part B. Part A contains 10 compulsory short answer questions, 2 from each module. Each question carries 3 marks. Part B contains 2 questions from each module of which student should answer any one. Each question can have a maximum of 2 sub-divisions and carry 6 marks.

Sample Course Level Assessment Questions

Course Outcome 1 (CO 1):

1. How to perform industrial espionage investigation?
2. Explain the various steps involved in cyber forensics investigation.
3. Identify real-time scenarios that is considered as company policy violation.

Course Outcome 2 (CO 2):

1. What are the advantages of proprietary evidence format over row format?
2. Differentiate static acquisitions from live acquisitions.
3. How to ensure the integrity of collected digital evidence? List the techniques employed to validate the collected evidence.

Course Outcome 3 (CO 3):

1. Describe the various activities involved while starting a windows operating system.
2. Explain the various file system organization used in windows operating system.
3. What is the importance of windows registry analysis in forensic investigation?
4. How to ensure the integrity of collected evidences?
5. Differentiate the forensics procedure in Linux and MacOS.

Course Outcome 4 (CO 4):

1. Explain the mobile forensics procedure in detail.
2. Illustrate the use of Wireshark packet analyser.
3. How the forensics acquisition method in mobile differs from that in computer system?

Course Outcome 5 (CO 5):

1. How to write an investigation report that can sustain in court of law?
2. Discuss how Autopsy tool is used to generate forensics report.
3. Address the difficulties that occurred while preparing an Expert Testimony.

Model Question Paper

Course Code: 20MCA267

Course Name: Cyber Forensics

Max. Marks :60 Duration: 3 Hrs

Part A

Answer all questions.

Each question carries 3 marks (10 x 3 = 30 Marks)

1. Categorize the formats used to store the collected digital evidences.
2. What do you mean by a computer crime? Which activities are considered as company policy violations?
3. Enumerate the features of Resilient File System.
4. Write down the operations involved in boot sequence.
5. Differentiate between soft link and hard link.
6. Which are the data acquisition tools available in Linux that is forensically sound?
7. List the features of Wireshark tool.

8. List different types of mobile forensic acquisition procedures.
9. State the guidelines for writing a report which is admissible in a court of law.
10. What are the different types of forensics reports?

(10 x 3=30 marks)

Part B

Answer all questions. Each question carries 6 marks. (5 * 6 = 30 Marks)

11. What is evidence bag? Describe standard operating procedures for securing evidence before transporting it to forensic lab. (6)

OR

12. How the retention policy of evidence related to evidence storage mediums? (6)

13. Explain the importance of Windows Registry in forensics analysis. (6)

OR

14. What is a solid-state storage device? Discuss the usage of Microsoft BitLocker tool. (6)

15. Explain the file structures of Linux and MacOS. (6)

OR

16. Define write blocker? Explain the use of Sleuth Kit tool. (6)

17. Explain the standard operating procedures used for mobile forensics. (6)

OR

18. Discuss the standard operating procedures used for network forensics. (6)

19. List and explain the steps involved in generating forensics report using Autopsy tool. (6)

OR

20. What are the responsibilities of a Computer Forensic Investigator? (6)

(5 x 6=30 Marks)

Syllabus

Module 1: (8 Hours)

Overview of computer crime, Overview of company policy violation, Preparing a case - Planning an investigation, Securing evidence. Industrial espionage investigation.

Conducting an investigation: Gathering evidence, Bit-stream copy of evidence.

Storage formats for storing collected digital evidence - Raw format, Proprietary formats, Advanced Forensic Format (AFF). Acquisition tools and methods. Digital evidence validation methods and tools.

Storing Digital evidence- Evidence Retention.

Familiarizing Autopsy for Windows - a free forensics tool.

Module 2: (10 Hours)

Understanding Digital data and storage systems: Understanding boot sequence, Understanding Disk Drives - Solid-state Storage Devices (SSDs).

Microsoft File Systems - Disk partitions, Understanding FAT, Understanding NTFS, MFT - file attributes, file data, NTFS compressed files, NTFS encrypted file system, Deleting NTFS file system, ReFS.

Whole disk encryption, Microsoft BitLocker. Understanding Windows Registry. Microsoft Windows startup tasks.

A practical assignment may be given in encrypting a partition of your computer hard disk drive/ encrypting USB flash drive to avoid firm-level attack.

Module 3: (10 Hours)

Linux file structures - File structures in Ext4, Hard links and Symbolic links.

Macintosh (MacOS) file structures - Forensic procedures in MacOS.

Setting up Sleuth Kit and Autopsy - Examining a case with Sleuth Kit and Autopsy, Importance of Write-blocker.

Acquiring data with a Linux boot CD - Preparing a target drive for data acquisition, Using dd and dcfldd commands.

Validating data acquisitions - Linux validation methods, Windows validation methods.

Following practical assignments may be given:

i. *Recover deleted files from pen drive*
ii. *Extract camera information from recovered images*
iii. *Extract deleted internet browsing history*
iv. *Recover deleted files from unallocated space using Autopsy*

Module 4: (10 Hours)

Understanding Mobile Device forensics - Mobile phone basics, Understanding Mobile phone hardware.

Acquisition procedures for Mobile devices, Mobile Forensic equipment, SIM card readers, Mobile phone Forensics tools and methods.

Network Forensics - The Need for Established Procedures, Securing a Network, Developing Procedures for Network Forensics, Wireshark packet analyser.

Practical assignments may be given:

i. *Identify students who use college lab facility to browse shopping websites*
ii. *Identify the hacking attempt on a closed port using ping sweep*
iii. *Using Wireshark retrieve the username and password of users who browse less secure website with Wi- Fi connection*

Module 5: (7 Hours)

Understand the importance of Forensics Reports, Types of reports, Guidelines for writing reports, Layout and presentation of reports, Generating reports with Autopsy.

Ethics and codes for Expert Witness - Forensics Examiner's role in testifying, Considerations in disqualification, Determining admissibility of evidence. Ethical difficulties in Expert Testimony, Ethical responsibilities.

Text Book

1. Bill Nelson, Amelia Phillips, Christopher Steuart, "Guide to Computer Forensics and Investigations", Cengage Learning, 6th Edition.

Reference Books

1. Marjie T. Britz, "Computer Forensics and Cyber Crime", Pearson Third Edition 2013.
2. Marie - Helen Maras "Computer Forensics: Cybercriminals, Laws, and Evidence", Jones & Bartlett Learning, Second Edition 2015.

Web References

1. https://www.wireshark.org/download/docs/user-guide.pdf (Reference for Wireshark)
2. http://www.open.edu/openlearn/futurelearn/cyber-security

Course Contents and Lecture Schedule

Sl. No.	Topic	No. of Lectures
1	**Module 1**	**8 Hours**
1.1	An overview of computer crimes and company policy violations	1
1.2	Preparing a case - Planning an investigation, Securing evidence. Industrial espionage investigation	1
1.3	Conducting an investigation: Gathering evidence, Bit-stream copy of evidence	1
1.4	Storage formats for storing collected digital evidence - Raw format, Proprietary formats, Advanced Forensic Format (AFF)	1
1.5	Acquisition tools and methods	1
1.6	Digital evidence validation methods and tools	1
1.7	Storing Digital evidence -Evidence Retention	1
1.8	Familiarizing Autopsy for Windows - a free forensics tool	1
2	**Module 2**	**10 Hours**
2.1	Understanding Digital data and storage systems, Understanding boot sequence	1
2.2	Understanding Disk Drives	1
2.3	Solid-state Storage Devices (SSDs)	1
2.4	Microsoft File Systems - Disk partitions	1
2.5	Understanding FAT	1
2.6	Understanding NTFS, MFT - file attributes, file data	1
2.7	NTFS compressed files, NTFS encrypted file system	1
2.8	Deleting NTFS file system, ReFS	1
2.9	Whole disk encryption, Microsoft BitLocker	1
2.10	Understanding Windows Registry, Microsoft Windows startup tasks	1
3	**Module 3**	**10 Hours**
3.1	Linux file structures - File structures in Ext4	1
3.2	Hard links and Symbolic links	1
3.3	Macintosh (MacOS) file structures - Forensic procedures in MacOS	1
3.4	Setting up Sleuth Kit and Autopsy - Examining a case with Sleuth Kit and Autopsy	1
3.5	Importance of Write-blocker	1

3.6	Acquiring data with a Linux boot CD	1
3.7	Preparing a target drive for data acquisition	1
3.8	Using dd and dcfldd commands	1
3.9	Validating data acquisitions - Linux validation methods	1
3.10	Windows validation methods	1
4	**Module 4**	**10 Hours**
4.1	Understanding Mobile Device forensics - Mobile phone basics	1
4.2	Understanding Mobile phone hardware	1
4.3	Acquisition procedures for Mobile devices	1
4.4	Mobile Forensic equipment	1
4.5	SIM card readers	1
4.6	Mobile phone Forensics tools and methods	1
4.7	Network Forensics - The Need for Established Procedures	1
4.8	Securing a Network	1
4.9	Developing Procedures for Network Forensics	1
4.10	Wireshark packet analyser	1
5	**Module 5**	**7 Hours**
5.1	Understand the importance of Forensics Reports, Types of reports	1
5.2	Guidelines for writing reports, Layout and presentation of reports	1
5.3	Generating reports with Autopsy	1
5.4	Ethics and codes for Expert Witness - Forensics Examiner's role in testifying	1
5.5	Considerations in disqualification, Determining admissibility of evidence	1
5.6	Ethical difficulties in Expert Testimony	1
5.7	Ethical responsibilities	1

CODE	COURSE NAME	CATEGORY	L	T	P	CREDIT
20MCA269	**COMPILER DESIGN**	**ELECTIVE**	**3**	**1**	**0**	**4**

Preamble: The objective of this course is to explore the principles, algorithms and data structures involved in the design of compilers. It includes lexical analysis, parsing techniques, generating grammars, intermediate code generation, code optimization and code generation phases.

Prerequisite: Nil

Course Outcomes: After completion of the course the student will be able to

CO No.	Course Outcome (CO)	Bloom's Category Level
CO 1	Explain different phases of compiler and perform lexical analysis using the concepts of regular expressions and finite automata.	Level 2: Understand
CO 2	Develop top down and bottom-up parsers to perform syntax analysis using context free grammar.	Level 3: Apply
CO 3	Explain syntax directed translation schemes and type checking for a given grammar.	Level 2: Understand
CO 4	Distinguish different intermediate code representations and generate intermediate code for statements in high level languages.	Level 2: Understand
CO 5	Describe various code optimization techniques and generate machine dependent code.	Level 2: Understand

Mapping of Course Outcomes with Program Outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	2	2		-	-	-	2	-	-	-	-	-
CO 2	2	2	2	-	-	-	2	-	-	-	-	-
CO 3	2	2		-	-	-	2	-	-	-	-	-
CO 4	2	1		-	-	-	2	-	-	-	-	-
CO 5	2	3		-	-	-	2	-	-	-	-	-

3/2/1: High/Medium/Low

Assessment Pattern

Bloom's Category Levels	Continuous Assessment Tests		End Semester Examination
	1	2	
Level 1: Remember	15	15	20
Level 2: Understand	25	35	30
Level 3: Apply	10		10
Level 4: Analyse			
Level 5: Evaluate			
Level 6: Create			

Mark distribution

Total Marks	Continuous Internal Evaluation (CIE)	End Semester Examination (ESE)	ESE Duration
100	40	60	3 hours

Continuous Internal Evaluation Pattern:

Attendance	: 8 marks
Continuous Assessment Test (2 numbers)	: 20 marks
Assignment/Quiz/Course project	: 12 marks

End Semester Examination Pattern: There will be *two* parts; **Part A** and **Part B**. Part A contain 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer *all* questions. Part B contains 2 questions from each module of which student should answer *any one*. Each question carries 6 marks.

Sample Course Level Assessment Questions

Course Outcome 1 (CO 1):

1. Explain how the regular expressions and finite automata are used for specification and recognition of tokens.
2. State the role of lexical analyser. Identify the lexemes and their corresponding tokens in the following statement: printf (“Simple Interest=%f\n”, si);
3. Draw the DFA for the regular expression (a | b) * (abb | a+ b).
4. Trace the output after each phase of the compiler for the assignment statement:
   a = b + c * 10, if variables given are of float type.

Course Outcome 2 (CO 2):

1. Find the LR (0) items for the grammar
   S->SS | a| €.
2. Show the steps involved in recursive descent parsing with backtracking for the string cad with the given grammar: S -> cAd, A -> ab | a
3. Construct the predictive parsing table for the following grammar:
   S -> (L) | a
   L -> L,S | S

Course Outcome 3 (CO 3):

1. Write the S-attributed SDD of a simple desk calculator and show annotated parse tree for the expression (3+4) *(5+6).
2. Explain bottom- up evaluation of S- attributed definitions.
3. Explain the specification of a simple type checker

Course Outcome 4 (CO 4):

1. Draw DAG for the expression (a/10 + (b -10)) *(a/10 + (b-10)). Also write the sequence of instructions used for the DAG construction.
2. Write the three-address code sequence for the statement x=y*z + y*-z. Also give its triple representation.
3. Write syntax directed definitions to construct syntax tree and three address code for assignment statements.

Course Outcome 5 (CO 5):

1. Using code generation algorithm generate code sequence for the expression x = (a – b) + (a + c).
2. With suitable example of a basic block, explain the code-improving transformations of a basic block.
3. Explain common sub expression elimination with an example.

Model Question Paper
Course Code: 20MCA269

Course Name: COMPILER DESIGN

Max. Marks :60 **Duration: 3 Hrs**

Part A

Answer all questions. Each question carries 3 marks (10 * 3 = 30 Marks)

1 State the role of lexical analyzer. Identify the lexemes and their corresponding tokens in the following statement: printf (“Simple Interest=%f\n”, si); (3)

2 Draw the transition diagram for the regular definition, (3)

relop → < | <= | = | <> | >= | >

3 Find the FIRST and FOLLOW of the non-terminals in the grammar (3)

S-> aABe

A-> Abc|b

B-> d

4 Demonstrate the identification of handles in operator precedence parsing? (3)

5 What is a Syntax Directed Definition? Show an example. (3)

6 Distinguish between synthesized and inherited attributes. (3)

7 Write the three-address code sequence for the statement x=y*z + y*-z. Also give its triple representation. (3)

8 Discuss about the getreg() function in code generator algorithm (3)

9 Identify any two issues in the design of a Code Generator. (3)

10 Explain common sub expression elimination with an example. (3)

Part B

Answer all questions. Each question carries 6 marks. (5 * 6 = 30 Marks)

11 Explain the working of different phases of a compiler. Illustrate with a source language statement. (6)

OR

12 Explain how the regular expressions and finite state automata are used for the specification and recognition of tokens? (6)

13 Construct LALR parse table for the grammar (6)

S->C
C->cC|d

OR

14 Design a recursive descent parser for the grammar (6)

E->E + T | T
T->T*F | F
F->(E) | id

15 Differentiate between S-attributed and L-attributed definitions with suitable examples (6)

OR

16 Write the Syntax Directed Definition for a simple type declaration and draw the annotated parse tree for the declaration float a, b, c. (6)

17 Explain the following and show an example for each. (6)

i). Three-address code iii). Triples

ii). Quadruples iv). Indirect triples

OR

18 Write Syntax Directed Definition to produce three-address code for Boolean expressions and obtain the three-address code for the statement given below: (6)

```
 while a < b do
            if c < d then
                  x = y + z
               else
                  x = y – z
```

19 Explain different code optimization techniques available in local and global optimizations? (6)

OR

20 Write the code generation algorithm. Using this algorithm generate code sequence for the expression x = (a – b) + (a + c). (6)

Syllabus

Module 1 (8 Hours)
Introduction to compilers: Analysis of the source program, Phases of a compiler, Grouping of phases Lexical analysis: role of lexical analyser, input buffering, specification of tokens, recognition of tokens, Deterministic and Non-Deterministic Finite automata, Regular expression to NFA and DFA
Module 2 (12 Hours)
Syntax analysis: Role of parser, Context free grammars Top down parsing: Recursive Descent parsing, Predictive parsing, LL(1) Grammars. Bottom-up parsing: Shift Reduce Parsing, Operator Precedence Parsing (concepts only), LR parsing – Constructing SLR parsing tables, Constructing Canonical LR parsing tables and Constructing LALR parsing tables.
Module 3 (8 Hours)
Syntax directed translation: Syntax directed definitions, Bottom-up evaluation of S-attributed definitions, L- attributed definitions, Top-down translation, Bottom-up evaluation of inherited attributes. Type Checking: Type systems, Specification of a simple type checker.
Module 4 (7 Hours)
Intermediate code generation: Graphical representations, Three address code - Quadruples - triples - Indirect triples, Assignment Statements, Boolean Expressions, Control flow statements
Module 5 (10 Hours)
Code Optimization: Principal sources of optimization, Optimization of Basic blocks, Global data flow analysis. Code generation: Issues in the design of a code generator, The target machine, Basic blocks and flow graphs, A simple code generator, Peephole optimization.

Note : Programming assignments using lexical analyser generator, using parser generator.

Text Books

1. Alfred V.Aho , Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman, Compilers – Principles, Techniques and Tools, Addison Wesley, 2nd Edition,2006.

Reference Books

1. V Raghavan- Principles of Compiler Design – Tata McGraw Hill, 2nd edition,2011
2. Jean Paul Tremblay and Sorenson., The Theory and Practice of Compiler Writing ,McGraw Hill,2nd Edition,2006
3. Nandini Prasad, Principles of compiler design, Elsevier, 2nd Edition,2012
4. Kenneth C. Louden, Compiler Construction-Principles and Practice, 2nd Edition, Cengage, 2010.
5. Keith Cooper and Linda Torczon, Engineering a Compiler, 2nd Edition, Elsevier, 2011

Course Contents and Lecture Schedule

Sl. No.	Topic	No. of Lectures
1	**Lexical Analysis**	**8 Hours**
1.1	Lexical Analysis: Analysis of the source program	1
1.2	Phases of a compiler, Grouping of phases	1
1.3	Lexical analysis: role of lexical analyser, input Buffering	1
1.4	specification of tokens, recognition of tokens	1
1.5	Deterministic and Non-Deterministic Finite automata	2
1.6	Regular expression to NFA and DFA	2
2	**Syntax Analysis**	**12 Hours**
2.1	Syntax analysis: Role of parser, Context free grammars	1
2.2	Top-down parsing: Recursive Descent parsing	2
2.3	Predictive parsing, LL(1) Grammars	2
2.4	Bottom-up parsing: Shift Reduce Parsing	1
2.5	Operator Precedence Parsing	1
2.6	LR parsing – Constructing SLR parsing tables	2
2.7	Constructing Canonical LR parsing tables	2
2.8	Constructing LALR parsing tables.	1

3	**Syntax directed translation and Type Checking**	**8 Hours**
3.1	Syntax directed translation: Syntax directed definitions	1
3.2	Bottom- up evaluation of S attributed definitions, L- attributed definitions	2
3.3	Top-down translation, Bottom-up evaluation of inherited attributes.	2
3.4	Type Checking: Type systems	1
3.5	Specification of a simple type checker.	2
4	**Intermediate code generation**	**7 Hours**
4.1	Intermediate code generation: Graphical representations	2
4.2	Three address code-quadruples -triples-Indirect triples	2
4.3	Assignment Statements, Boolean Expressions, Control flow statements	2
4.4	Control flow statements	1
5	**Code Optimization and Code Generation**	**10 Hours**
5.1	Code Optimization: Principal sources of optimization	2
5.2	Optimization of Basic blocks,	1
5.3	Global data flow analysis	2
5.4	Code generation: Issues in the design of a code generator.	2
5.5	The target machine, Basic blocks and Flow graphs	2
5.6	Peephole optimization	1

CODE	COURSE NAME	CATEGORY	L	T	P	CREDIT
20MCA281	INTERNET OF THINGS	ELECTIVE	3	1	0	4

Preamble: This course intends to provide insight into new innovations that will build novel type of interactions among things and humans, and enables the realization of smart cities, infrastructures, and services for enhancing the quality of life and utilization of resources. An overview of IOT and its related concepts, different IOT architectures and their components, emerging paradigms such as Fog computing, Platforms and solutions supporting development and deployment of IOT applications, message passing mechanisms such as RPC, REST, and CoAP, data and knowledge management, data confidentiality, data integrity, and operation control issues faced by IOT are included in the course.

Prerequisite: Basic concepts of Information Technology and Internet.

Course Outcomes: After completion of the course the student will be able to

CO No:	Course Outcome (CO)	Blooms Category Level
CO 1	Describe the main concepts and features of the IOT paradigm.	Level 2: Understand
CO 2	Discuss Fog computing, TinyOS - nesC and programming frameworks for IOT	Level 2: Understand
CO 3	Describe the data management techniques applied to the IOT environment.	Level 2 Understand
CO 4	Explain security, and privacy in IOT environments	Level 2 Understand
CO 5	Discuss key enablers and solutions to enable practical IoT systems	Level 2 Understand

Mapping of course outcomes with program outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	3						2					
CO 2	3	1					2					
CO 3	3	1					2					
CO 4	3	1					2					
CO 5	3	1	1				2					

3/2/1: High/Medium/Low

Assessment Pattern

Bloom's Category	**Continuous Assessment Tests**		**End Semester Examination**
	1	**2**	
Remember	20	20	20
Understand	30	30	40

Apply			
Analyse			
Evaluate			
Create			

Mark Distribution

Total Marks	**CIE**	**ESE**	**ESE Duration**
100	40	60	3 hours

Continuous Internal Evaluation Pattern:

Attendance	: 8 marks
Continuous Assessment Test (2 numbers)	: 20 marks
Assignment/Quiz/Course project	: 12 marks

End Semester Examination Pattern: There will be two parts; Part A and Part B. Part A contain 10 compulsory short answer questions, 2 from each module. Each question carries 3 marks. Part B contains 2 questions from each module of which student should answer any one. Each question can have a maximum of 2 subdivisions and carry 6 marks.

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Compare SOA-based architecture and API-oriented architecture.
2. Neatly sketch the open IOT architecture for IOT/CLOUD convergence.
3. List and explain the applications of device/cloud collaboration.

Course Outcome 2 (CO2)

1. What are the advantages associated with Fog computing?
2. Comment on the four broad requirements that motivate the design of TinyOS.
3. Summarize the communication paradigms and technologies used in resource-constrained environments.

Course Outcome 3(CO3):

1. Explain stream and stream processing in IOT.
2. Write and explain the algorithm for distributed anomaly detection by clustering ellipsoids.
3. Discuss the general architecture of a stream-processing system in IOT.

Course Outcome 4 (CO4):

1. Give an overview on the security requirements of IOT.
2. How can you nullify the impact of fault in high-availability cluster?
3. Explain the BCK with pre-shared keys for TinyTO.

Course Outcome 5 (CO5):

1. Give an overview on the Wired Gateway Interfaces.
2. List the features to select the gateway hardware.
3. List the steps to prepare Raspberry Pi for the execution.

Model Question Paper
Course Code: 20MCA281

Course Name: INTERNET OF THINGS

Max. Marks :60 **Duration: 3 Hrs**

Part A

Answer all questions. Each question carries 3 marks (10 * 3 = 30 Marks)

1. What do you mean by computation offloading?
2. Explain the framework that enables collaboration between smart mobile devices and cloud.
3. Outline the major challenges faced in the Fog paradigm.
4. Explain Polyglot Programming.
5. Which are the challenges faced by stream-processing systems?
6. Explain anomaly detection and categorize anomalies in the data.
7. List the different ways that an IOT gateway can extend connectivity to nodes.
8. Write the advantages of obfuscation and diversification techniques.
9. Explain Inter-Integrated Circuit (I^2C) or Two Wire Interface (TWI).
10. Write a short note on Zigbee.

Part B

Answer all questions. Each question carries 6 marks. (5 * 6 = 30 Marks)

11. Explain the taxonomy of Resource Management in IOT. (6 Marks)

OR

12. Draw and explain the state diagram of the open IOT services life cycle. (6 Marks)

13. a. Comment on the four broad requirements that motivate the design of TinyOS (3 Marks)

    b. Describe the design decisions for nesC. (3 Marks)

OR

14. List the features in coordination languages - Linda, eLinda, Orc, and Jolie (6 Marks)

15. Compare Stream Management System (DSMS) and Complex Event Processing (CEP). (6 Marks)

OR

16. Describe hyper ellipsoidal model for anomaly detection. (6 Marks)

17. Describe the error detection techniques which are applicable in the context of an IOT. (6 Marks)

OR

18. Explain the Station-to-Station protocol (STS) and the two main shortcomings of STS.

(6 Marks)

19. Discuss the sensors required to build the environmental-sensing IoT gateway device for weather monitoring. (6 Marks)

OR

20. List and explain the six steps for the development of a sensor project. (6 Marks)

Syllabus

Module 1 (9 Hours)
Overview of Internet of Things: Open-source semantic web infrastructure for managing IOT resources in the Cloud - Device/Cloud Collaboration framework for intelligence applications.
Module 2 (11 Hours)
Introduction to Fog Computing: principles, architectures, and applications. TinyOS – NesC, Programming frameworks for Internet of Things
Module 3 (8 Hours)
Stream processing in IoT: foundations, state-of-the-art, and future directions - A framework for distributed data analysis for IoT
Module 4 (9 Hours)
Security and privacy in the Internet of Things- Internet of Things - robustness and reliability. TinyTO: two-way authentication for constrained devices in the Internet of Things - Obfuscation and diversification for securing the Internet of Things

Module 5 (8 Hours)
Creating a simple IoT project - Preparing Raspberry Pi – Interfacing the hardware - Internal representation of sensor values- Persisting data - Creating the actuator project - Creating a controller.

More detailed knowledge may be acquired through seminars, assignments and talks by eminent external experts and also by implementing a micro project.

Any one of the following or similar micro projects may be given as part of the course.

1. Smart Gas Leakage Detector
2. Night Patrol at home

Text Books

1. RajkumarBuyya; Amir VahidDastjerdi , “Internet of Things”, Morgan Kaufmann, 2016

Reference Books

1. Peter Waher, “Learning Internet of Things”, Packt Publishing, 2015
2. S. SitharamaIyengar; Nandan Parameswaran; Vir V. Phoha; N. Balakrishnan; Chuka Okoye, “Fundamentals of Sensor Network Programming: Applications and Technology”, Wiley, December 14, 2010
3. Robert Stackowiak, Art Licht, VenuMantha, Louis Nagode, “Big Data and The Internet of Things: Enterprise Information Architecture for A New Age”, Apress, 2015

Web Resources

1. https://www.coursera.org/specializations/internet-of-things
2. http://web.mit.edu/professional/digital-programs/courses/IoT

Course Contents and Lecture Schedule

No	Topic	No. of Lectures
1	**Introduction (9 Hours)**	
1.1	Internet of things- definition, evolution. Applications -Smart home applications, Health care, Elder care, Traffic surveillance.	1
	SOA -Based Architecture, API oriented Architecture, Resource Management.	1
	Computational Offloading, Identification and Resource/Service Discovery, IOT Data Management and Analytics, IOT and the CLOUD	1

1.2	Open IOT architecture for IOT/Cloud convergence, Sensor middleware, Cloud computing infrastructure, Directory service, Global Scheduler, Local Scheduler component, Service delivery and utility manager	1
	Workflow of open IOT platform, Scheduling process and IOT Services lifecycle, State diagram of the Open IOT Services lifecycle within the scheduler module	1
	Scheduling and resource management, Resource optimization schemes, Caching technique	1
	Service creation flowchart, Comparison of cost - with cache server and public cloud data-score	1
1.3	Runtime adaptation engine, Device/cloud collaboration framework	1
	applications of device/cloud collaboration, Semantic QA cache	1
2	**Programming frameworks (11 Hours)**	
2.1	Introduction to Fog Computing: principles, architectures, and Applications	1
	Motivating scenario for Fog Computing, Advantages of Fog Computing, Reference architecture of Fog Computing	1
	Software-Defined Resource management layer, Services of Software-Defined Resource management layer, Applications of Fog Computing.	1
2.2	History of TinyOS, Implementation, Requirements motivating the design of TinyOS, Component Model, Interfaces. TinyOS computational concepts	1
	Overview of TinyOS Execution Model, Concurrency, TinyOS Theory of Execution: Events & Tasks, TinyOS Architecture. TinyOS-Programming Model.	1
2.3	nesC design, Component Implementation, Design Decisions for nesC, Module Components, Configuration Components	1
	Whole-Program Analysis, Detecting Race Conditions, Dealing with Race Conditions, Issues for nesC.	1
2.4	Overview of Embedded Programming Languages- nesC, Keil C, Dynamic C, B#, Message Passing in Devices-Remote Procedure Call (RPC), Lightweight RPC (LRPC)	1
	Representational state transfer (REST), Computational REST (CREST), Constrained Application Protocol(CoAP), Comparison of HTTP and CoAP, Advantages of CoAP	1

	Coordination Languages- Orchestration, Choreography, Linda and eLinda, Orc, Features of Orc, Java Orchestration Language Interpreter Engine (Jolie), Polyglot Programming, Inverse pyramid for Polyglot Programming.	1
	Features of programming frameworks for IOT, IOT programming approaches, Existing IOT frameworks	1
3	**Data management techniques (8 Hours)**	
3.1	Stream, Stream Processing, Data Stream Management System (DSMS)	1
	Complex Event Processing (CEP), differences between two use-cases of Stream Processing: DSMS and CEP	1
	The characteristics of stream data in IOT, general architecture of a stream-processing system in IOT	1
	Continuous logic processing system, challenges in stream-processing systems.	1
3.2	Anomaly detection, problem statement and definitions	1
	Hyper ellipsoidal anomaly detection	1
	Distributed anomaly detection	1
	Clustering ellipsoids, incremental local modeling	1
4	**Security and privacy (9 Hours)**	
4.1	IOT security threats, IOT security requirements, security frameworks for IOT, IOT security overview, IOT gateways and security, IOT routing attacks	1
	Security frameworks for IOT - Lightweight cryptography, asymmetric LWC algorithms, privacy in IOT networks	1
4.2	IOT characteristics and reliability issues, reliability challenges	1
	Addressing reliability, security aspects and solutions	1
4.3	TinyTO: Two-way authentication for constrained devices in the Internet of Things	1
	TinyTO protocol	1
	BCK with pre-shared keys for TinyTO, handshake implementation	1
4.4	IOT network stack and access protocols, Obfuscation and diversification techniques	1

	Enhancing the security in IOT using obfuscation and diversification techniques, motivations and limitations, different use-case scenarios on software diversification and obfuscation.	1
5	**IoT Implementation (8 Hours)**	
5.1	Three key components to an IOT architecture, Sensor to gateway communication - wired gateway interfaces, wireless gateway interfaces	1
	Sensors - sensors required to build the environmental-sensing IOT gateway device for weather monitoring	1
	Gateway, Gateway hardware, Gateway software	1
	Data transmission - advanced message queuing protocol, backend processing, to CLOUD or not to cloud	1
5.2	Creating a simple sensor project - Preparing Raspberry Pi – Clayster libraries	1
	Hardware, Interfacing the hardware - Internal representation of sensor values- Persisting data	1
	External representation of sensor values, Exporting sensor data	1
	Creating the actuator project – Hardware, Interfacing the hardware, Creating a controller	1

20MCA283	DEEP LEARNING	CATEGORY	L	T	P	CREDIT
		ELECTIVE	3	1	0	4

Preamble: This course intends to provide insight into deep learning. This topic is currently a much sought-after skill and is under active research. Students have to refer appropriate research papers and multiple books to get in-depth knowledge about the topics**.** Instructors may give suitable programming assignments to augment the material covered in the classroom.

Prerequisite: Basic concepts of linear algebra, probability and optimization.

Course Outcomes: After the completion of the course the student will be able to

CO No.	Course Outcome (CO)	Bloom's Category Level
CO 1	Explain the basic concepts of deep learning.	Level 2: Understand
CO 2	Design neural networks using TensorFlow	Level 3: Apply
CO 3	Solve real world problems with CNN.	Level 3: Apply
CO 4	Solve real world problems with RNN.	Level 3: Apply
CO 5	Describe the concepts of GAN.	Level 2: Understand

Mapping of course outcomes with program outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	2	2										
CO 2	3	3	3		3		3					
CO 3	3	3	3		3		3					
CO 4	3	3	3		3		3					
CO 5	2	3			2		2					

3/2/1: High/Medium/Low

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination
	1	2	
Remember	15	15	10
Understand	25	25	30
Apply	10	10	20
Analyze			
Evaluate			
Create			

Mark distribution

Total Marks	CIE	ESE	ESE Duration
100	40	60	3 hours

Continuous Internal Evaluation Pattern:

Attendance	: 8 marks
Continuous Assessment Test (2 numbers)	: 20 marks
Assignment/Quiz/Course project	: 12 marks

End Semester Examination Pattern: There will be two parts; Part A and Part B. Part A contains 10 compulsory short answer questions, 2 from each module. Each question carries 3 marks. Part B contains 2 questions from each module of which student should answer any one. Each question can have maximum 2 sub-divisions and carry 6 marks

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Describe the model of a biological neuron.
2. Explain Perceptron learning algorithm.
3. Explain the role of batch normalization in training a neural network.

Course Outcome 2 (CO2)

1. Draw and demonstrate the VGG-16 architecture.
2. Sketch the AlexNet architecture and explain its functionalities.

Course Outcome 3(CO3):

1. Design a convolutional neural network which can classify MNIST handwritten data.
2. An input image has been converted into a matrix of size 12 X 12 along with a filter of size 3 X 3 with a Stride of 1. Determine the size of the convoluted matrix.
3. Why do we prefer Convolutional Neural networks (CNN) over Artificial Neural networks (ANN) for image data as input?

Course Outcome 4 (CO4):

1. You are given an image data set with 10 classes. Describe how you will use deep learning to build a classifier.
2. Design a system to generate deep fakes from an image.

Course Outcome 5 (CO5):

1. Describe auto encoders and how they help in dimensionality reduction.
2. Explain how GANS work.

Model Question Paper
Course Code: 20MCA283

Course Name: DEEP LEARNING

Max. Marks :60 **Duration: 3 Hrs**

Part A

Answer all questions. Each question carries 3 marks (10 * 3 = 30 Marks)

1. Describe sigmoid activation functions.
2. Write the gradient descent algorithm.
3. Explain with an example how graphs are stored and represented in TensorFlow.
4. Discuss how graph representation can accelerate computing models.
5. Describe the VGG 16 architecture.
6. What is max pooling in the context of CNN?
7. Explain ReLU.
8. Explain the problem of vanishing gradients.
9. Write a note on auto encoders.
10. Explain the idea behind cross entropy.

Part B

Answer one full question from each module, each carries 6 marks.

11. (a) Describe the model of a biological neuron. 3 marks

(b) Explain perceptron learning algorithm. 3 marks

OR

12. With a suitable example explain how backpropagation works 6marks

13. Explain the role of batch normalization in training a neural network and describe how to find out overfitting from training and validation curves 6 marks

OR

14. Explain the ideas of Rank, Shape and Type with an example in the context of a Tensor Data Structure 6 marks

15. With a suitable numerical example illustrate convolution operation. 6 marks

OR

16. Explain the architecture of AlexNet. 6 marks

17. Explain the idea of Truncated backpropagation through time. 6 marks

OR

18. Describe how LSTM works. 6 marks

19. Distinguish between generative and discriminative models 6 marks

OR

20. Explain how a GAN is trained. 6 marks

Syllabus

Module I (8 Hours)
Review of Neural Networks: Model of a biological neuron, McCulloch Pitts Neuron, Activation Functions, Perceptron, Perceptron Learning Algorithm and Convergence, Multilayer Perceptron, Back propagation, Learning XOR, Sigmoid Neurons, Gradient Descent, Feed forward Neural Networks.
Module II (10 Hours)
Training Neural Networks: Initialization, dropout, batch normalization and dropout, overfitting, underfitting, training and validation curves. Data Visualization: Feature and weight visualization, tSNE. Introduction to TensorFlow: graphs, nodes, Tensor data structures - rank, shape, type, Building neural networks with TensorFlow, Introduction to Keras.
Module III (10 Hours)
Convolutional Neural Networks: Convolution operation, Convolutional layers in neural network, pooling, fully connected layers. Case study: Architecture of Lenet, Alexnet and VGG 16
Module IV (8 Hours)
Recurrent Neural Networks: Back propagation, vanishing gradients, exploding gradients, truncated backpropagation through time, Gated Recurrent Units (GRUs), Long Short-Term Memory (LSTM) cells, solving the vanishing gradient problem with LSTMs.
Module V (9 Hours)
Autoencoders, variational autoencoders. Generative Adversarial Networks (GAN): Discriminative and generative models, GAN discriminator, GAN generator, upsampling, GAN Training, GAN challenges, loss functions, cross entropy, minimax loss, Wasserstein loss.

Programming assignments using TensorFlow maybe given at the end of each module to get hands on experience.

Textbooks.

1. Generative Deep Learning: David Foster, OReily, (2019)
2. Deep Learning, Ian Goodfellow, Yoshua Bengio and Aaron Courville, MIT press (2016)
3. Hands on Machine Learning with Scikit Learn and TensorFlow, Aurélien Géron (2019)
4. Deep Learning Illustrated, Jon Krohn, Grant Beyleveld, Aglae Bassens, Pearson, 1st Edn., (2020)
5. Online book Dive Deep into Machine Learning at https://d2l.ai/

References

Module 1

a. https://www.cse.iitm.ac.in/~miteshk/CS6910/Slides/Lecture2.pdf
b. https://www.cse.iitm.ac.in/~miteshk/CS6910/Slides/Lecture3.pdf

Module 2

a. http://neuralnetworksanddeeplearning.com
b. Hands on Machine Learning with Scikit Learn and TensorFlow, Aurélien Géron
c. Probabilistic Machine Learning: An Introduction, Kevin Murphy
d. https://www.researchgate.net/publication/228339739_Viualizing_data_using_t-SNE

Module 3

a. https://www.cse.iitm.ac.in/~miteshk/CS7015/Slides/Teaching/pdf/Lecture11.pdf
b. Convolutional neural networks for visual computing (Chapter 4), Ragav Venkatesan and Baoxin Li CRC press

Module 4

a. On the difficulty of training RNNs: https://arxiv.org/pdf/1211.5063.pdf
b. LSTM: A Search Space Odyssey: https://arxiv.org/abs/1503.04069
c. Understanding Deriving and Extending the LSTM: https://r2rt.com/written-memories-understanding-deriving-and-extending-the-lstm.html
d. Understanding LSTM Networks: http://colah.github.io/posts/2015-08-Understanding-LSTMs/
e. https://www.cse.iitm.ac.in/~miteshk/CS7015/Slides/Teaching/pdf/Lecture14.pdf
f. https://www.cse.iitm.ac.in/~miteshk/CS7015/Slides/Teaching/pdf/Lecture15.pdf

Module 5

a. GANs in Action: Deep Learning with Generative Adversarial Network Jakub Langgr, Vladimir Bok
b. Generative Deep Learning: Teaching Machines to Paint, Write, Compose, and Play David Foster
c. https://developers.google.com/machine-learning/gan

Course Contents and Lecture Schedule

No	Topic	No. of Lectures
1	**Module 1**	**8 Hours**
1.1	Review of Neural Networks: Model of a biological neuron	1
1.2	McCulloch Pitts Neuron, Activation functions	1
1.3	Perceptron, Perceptron Learning Algorithm	1
1.4	Convergence, Multilayer Perceptron	1
1.5	Back propagation	1
1.6	Learning XOR, Sigmoid Neurons	1
1.7	Gradient Descent, Feed forward Neural Networks	2
2	**Module 2**	**10 Hours**
2.1	Training Neural Networks	1
2.2	Initialization, Dropout	1
2.3	Batch normalization and drop out	1
2.4	Over fitting, under fitting, training and validation curves, data visualization, feature and weight visualization, tSNE	2
2.5	Introduction to TensorFlow, graphs, nodes, Tensor Data Structures - rank, shape, type	2
2.6	Building neural networks with tensor flow	2
2.7	Introduction to Keras	1
3	**Module 3**	**10 Hours**
3.1	Convolutional neural networks	1
3.2	Convolution operation	2
3.3	Back propagation in multilayer neural networks	1
3.4	Convolutional layers in neural network, pooling	2
3.5	Fully connected layers	2
3.6	Case study: Architecture of Lenet, Alexnet and VGG 16	2
4	**Module 4**	**8 Hours**
4.1	Recurrent neural networks	1
4.2	Back propagation: vanishing gradients, exploding gradients	1
4.3	Truncated Backpropagation Through Time	1
4.4	LSTM	1
4.5	Gated Recurrent Units (GRUs)	1
4.6	Long Short-Term Memory (LSTM) Cells	1
4.7	Solving the vanishing gradient problem with LSTMs	2
5	**Module 5**	**9 Hours**
5.1	Autoencoders, Variational autoencoders	2
5.2	Generative Adversarial Networks (GAN)	1
5.3	Discriminative and generative models	2
5.4	GAN Discriminator, GAN Generator, upsampling,	1
5.5	GAN Training	1
5.6	GAN challenges, Loss functions, cross entropy, minimax loss, Wasserstein loss	2

CODE	COURSE NAME	CATEGORY	L	T	P	CREDIT
20MCA285	DIGITAL IMAGE PROCESSING	ELECTIVE	3	1	0	4

Preamble: This course introduces the techniques of simulating human vision into computer vision based on feature extraction to develop applications in different areas. The concept of enhancement, transforms, smoothing, restoration, compression, morphological image analysis, classification & segmentation in two-dimensional space are introduced. This course serves as a prerequisite for many advanced courses in computer vision areas.

Prerequisite: Nil

Course Outcomes: After completion of the course the student will be able to

CO No.	Course Outcome (CO)	Bloom's Category Level
CO 1	Discuss the fundamental concepts of digital image processing, image formation and representation of images.	Level 2: Understand
CO 2	Summarise image enhancement methods in the spatial domain.	Level 2: Understand
CO 3	Explain image transforms and image smoothing & sharpening using various kinds of filters in frequency domain.	Level 2: Understand
CO 4	Describe various methods in image restoration and compression.	Level 2: Understand
CO 5	Discuss morphological basics and image segmentation methods.	Level 2: Understand

Mapping of Course Outcomes with Program Outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	2	2					2					
CO 2	2	2					2					
CO 3	2	2					2					
CO 4	2	2					2					
CO 5	2	2					2					

3/2/1: High/Medium/Low

Assessment Pattern

Bloom's Category Levels	Continuous Assessment Tests		End Semester Examination
	1	**2**	
Level 1: Remember	15	15	20
Level 2: Understand	35	35	40
Level 3: Apply			
Level 4: Analyse			
Level 5: Evaluate			
Level 6: Create			

Mark distribution

Total Marks	Continuous Internal Evaluation (CIE)	End Semester Examination (ESE)	ESE Duration
100	40	60	3 hours

Continuous Internal Evaluation Pattern:

Attendance	: 8 marks
Continuous Assessment Test (2 numbers)	: 20 marks
Assignment/Quiz/Course project	: 12 marks

End Semester Examination Pattern: There will be two parts; Part A and Part B. Part A contains 10 compulsory short answer questions, 2 from each module. Each question carries 3 marks. Part B contains 2 questions from each module of which student should answer any one. Each question can have a maximum of 2 sub-divisions and carry 6 marks.

Sample Course Level Assessment Questions

Course Outcome 1 (CO 1):

1. List out various components of an Image Processing System.
2. Define Electromagnetic Spectrum.
3. Illustrate the image formation in the eye. Calculate the size of the retinal image of a tree, if the observer is looking at a tree 20 m high at a distance of 100.

Course Outcome 2 (CO 2):

4. Describe the basic relationships and distance measures between pixels in a digital image.
5. List and explain steps in Histogram Processing.
6. List and explain various Intensity transformation functions used in grey scale images.

7. Explain the process of Unsharp masking?

Course Outcome 3 (CO 3):

1. Explain the properties of Unitary transform.
2. Compare and contrast 1D-DFT and 2D-DFT.
3. Design a basic Laplacian filter using first order and second order derivatives.
4. Describe various image smoothing techniques using frequency domain filters.

Course Outcome 4 (CO 4):

1. Explain image noise models and list out different noise probability density functions used in image processing applications.
2. Describe Wiener filtering technique.
3. Draw the functional block diagram of image compression system. List various types of redundancy in compression?

Course Outcome 5 (CO 5):

1. Differentiate erosion and dilation in morphological processing.
2. Compare Global thresholding and Otsu's method.
3. Explain how does Hough transform works.

Model Question Paper

Course Code: 20MCA285

Course Name: Digital Image Processing

Max. Marks :60 **Duration: 3 Hrs**

Part A

Answer all questions.

Each question carries 3 marks (10 x 3 = 30 Marks)

.

1. Describe the elements of visual perception.
2. Define Toeplitz & Circulant matrices
3. Explain histogram equalization in detail.
4. Differentiate linear spatial filter and non-linear spatial filter.
5. Explain the properties of 2D DFT.
6. List the steps involved in frequency domain filtering.
7. Write note on Point Spread Function.
8. List the components of a compression system.
9. Compare opening and closing in morphological processing of images.
10. Explain the merits and demerits of edge thresholding in segmentation.

(10 x 3=30 marks)

Part B

Answer all questions. Each question carries 6 marks. (5 * 6 = 30 Marks)

11. Explain fundamental steps in Digital Image Processing. (6)

OR

12. Differentiate sampling and quantization in image processing. (6)

13. Explain basic grey level transformation in spatial domain. (6)

OR

14. Compare Unsharp masking and High-boost filtering in Spatial filtering (6)

15. Explain Discrete Cosine Transform and its properties. (6)

OR

16. Explain the working of Homomorphic filtering with an example. (6)

17. Explain image restoration process in detail. (6)

OR

18. Differentiate lossy and lossless image compression methods (6)

19. Compare erosion and dilation in Morphological image analysis (6)

OR

20. Explain canny edge detector in detail (6)

(5 x 6=30 Marks)

Syllabus

Module 1: Overview of Digital Image Processing (9 Hours)

Digital Image Processing: Basic concepts, Difference between image processing and computer vision, Components of an image processing system. Image processing applications.

Mathematical preliminaries: Basic Vector and Matrix operations, Toeplitz, Circulant, Unitary & Orthogonal matrices.

Elements of Visual Perception: Structure of the human eye and image formation, Brightness adaptation and discrimination. Types of Images: Binary, Gray scale and Color Images. Image Sampling and Quantization: Digital image as a 2D array, Spatial and Intensity resolution, 2D-sampling theorem. RGB and HSI color models.

Module 2: Concept of Image enhancement & Spatial filtering (10 Hours)

Concept of Image enhancement, Basic grey level transformation functions: Image negative, Log transformation, Power-law transformation, Piecewise linear transformations. Histogram of an Image, Histogram equalization with illustration.

Fundamentals of Spatial Filtering: Mechanics of Spatial filtering, 2D correlation and convolution.

Smoothing spatial filters: Linear and Nonlinear types.

Sharpening spatial filters: Laplacian operator, Unsharp masking and High-boost filtering, Gradient based operators for image sharpening.

Module 3: Image Transform & Filtering in frequency domain (8 Hours)

Image Transform-representation of an image in frequency domain, Unitary transformation of an Image-transform pair equations in matrix form, Properties of unitary transforms. 1D-DFT, 2D-DFT of an image- Properties of 2D-DFT. DCT and its properties, Filtering an Image in the Frequency Domain– Steps of frequency domain filtering. Basic concept and illustration of frequency domain image smoothing and sharpening.

Module 4: Image Restoration & Compression (8 Hours)

Image Restoration: Concept of Image restoration, A Model of the Image Degradation/Restoration Process, Image Noise Models, Point Spread Function, Restoration using Inverse filtering, Wiener filtering.

Image compression: Need for compression, redundancy, classification of image compression schemes, A general image compression system, Huffman coding, Transform based compression, JPEG standard, Digital image watermarking-basic concept.

Module 5: Basics of morphological image processing & image segmentation (10 Hours)

Morphological image processing basics: erosion and dilation, opening and closing, Hit or Miss transformation.

Image segmentation: Fundamentals, Point detection, Line detection, Basic steps of edge detection, Hough transform, Edge detectors - Marr-Hildreth edge detector & Canny edge detector. Thresholding: Basics of intensity thresholding, Global thresholding and Otsu's method. Region-based segmentation: Region growing, Region Splitting and Merging.

Text Books

1. Rafael C., Gonzalez & Woods R.E., "Digital Image Processing", Pearson Education.

2. Jain A.K, "Fundamentals of Digital Image Processing", Prentice Hall, Eaglewood Cliffs, NJ.

Reference Books

1.Schalkoff R. J., "Digital Image Processing and Computer Vision", John Wiley 5. Pratt W.K., "Digital Image Processing", John Wiley

2.Al Bovick , "Handbook of Image and Video Processing" , Academic Press, 2000

Course Contents and Lecture Schedule

Sl. No.	Topic	No. of Lectures
1	**Overview of Data Image Processing**	**9 Hours**
1.1	Basic concepts of digital image processing, Image processing application	2
1.2	Mathematical preliminaries: Basic Vector and Matrix operations	1
1.3	Toeplitz, Circulant, Unitary & Orthogonal matrices	1
1.4	Elements of Visual Perception, Structure of human eye	1
1.5	Brightness adaptation and discrimination, Types of Images	1
1.6	Sampling and Quantization	1
1.7	Spatial and Intensity resolution, 2D-sampling theorem.	1
1.8	RGB and HSI color models.	1
2	**Concept of Image enhancement & Spatial filtering**	**10 Hours**
2.1	Concepts of Image enhancement, Basic grey level transformation functions: Image negative, Log transformation, Power-law transformation, Piecewise linear transformations.	2
2.2	Histogram of an Image, Histogram equalization with illustration	1

2.3	Fundamentals of Spatial Filtering: Mechanics of Spatial filtering	1
2.4	2D correlation and convolution	1
2.5	Smoothing spatial filters: Linear and Nonlinear types	1
2.6	Sharpening spatial filters: Laplacian operator	1
2.7	Unsharp masking	1
2.8	High-boost filtering	1
2.9	Gradient based operators for image sharpening	1
3	**Image Transform & Filtering in frequency domain**	**8 Hours**
3.1	Image Transform-representation of an image in the transform domain.	1
3.2	Unitary transformation of an Image, Properties of unitary transforms	1
3.3	1D-DFT	1
3.4	2D-DFT and its properties	1
3.5	DCT and its properties	1
3.6	Filtering an Image in the Frequency Domain– Steps of frequency domain filtering	1
3.7	Image smoothing using frequency domain filters – Ideal Lowpass Filters, Butterworth Lowpass filter & Gaussian Lowpass filter.	1
3.8	Image sharpening using frequency domain filters – Ideal Highpass Filters, Butterworth Highpass filter & Gaussian Highpass filter.	1
4	**Image Restoration & Image compression**	**8 Hours**
4.1	Image Restoration: Concept of Image restoration	1
4.2	A Model of the Image Degradation/Restoration Process	1
4.3	Image Noise Models	1
4.4	Point Spread Function	1
4.5	Restoration using Inverse filtering, Wiener filtering	1
4.6	Image compression: Need for compression, redundancy, classification of image compression schemes	1
4.7	A general image compression system Huffman coding, Transform based compression.	1
4.8	JPEG standard, Digital image watermarking-basic concept.	1
5	**Basics of morphological image processing & image segmentation**	**10 Hours**
5.1	Morphological image processing basics: erosion and dilation, opening and closing, Hit or Miss transformation.	1
5.2	Image segmentation: Fundamentals, Point detection, Line detection	1
5.3	Basic steps of edge detection - Hough transform	2
5.4	Edge detectors: Marr-Hildreth edge detector	1

5.5	Canny edge detector	1
5.6	Thresholding: Basics of intensity thresholding.	1
5.7	Global thresholding	1
5.8	Otsu’s method	1
5.9	Region-based segmentation: Region growing, Region Splitting and Merging.	1

CODE	COURSE NAME	CATEGORY	L	T	P	CREDIT
S20MCA287	**BIOINFORMATICS**	**ELECTIVE**	**3**	**1**	**0**	**4**

Preamble: This course helps to understand the concepts of computational biology and bioinformatics. The students will learn Database tools and their uses, various algorithms for biological sequence analysis, Genomics and Gene Recognition, Protein structure and to use various visualization techniques, data mining & machine learning in bioinformatics.

Prerequisite: Nil

Course Outcomes: After completion of the course the student will be able to

CO No.	Course Outcome (CO)	Bloom's Category Level
CO 1	Explain the fundamentals of Computational Biology and Bioinformatics.	Level 2: Understand
CO 2	Classify various biological databases.	Level 2: Understand
CO 3	Use suitable algorithm for Biological Sequence Analysis and make use of database search tools.	Level 3: Apply
CO 4	Discuss Gene structure and expression of Prokaryotic and Eukaryotes.	Level 2: Understand
CO 5	Apply data mining & machine learning methods to analyse and visualize biological data.	Level 3: Apply

Mapping of Course Outcomes with Program Outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	3	1	1				1					
CO 2	3	3	1				2					
CO 3	3	3	2				2					
CO 4	3	2	1				1					
CO 5	3	3	2		2		2					

3/2/1: High/Medium/Low

Assessment Pattern

Bloom's Category Levels	Continuous Assessment Tests		End Semester Examination
	1	2	
Level 1: Remember	10	10	10
Level 2: Understand	30	30	30
Level 3: Apply	10	10	20
Level 4: Analyse			
Level 5: Evaluate			
Level 6: Create			

Mark distribution

Total Marks	Continuous Internal Evaluation (CIE)	End Semester Examination (ESE)	ESE Duration
100	40	60	3 hours

Continuous Internal Evaluation Pattern:

Attendance	: 8 marks
Continuous Assessment Test (2 numbers)	: 20 marks
Assignment/Quiz/Course project	: 12 marks

End Semester Examination Pattern: There will be *two* parts; **Part A** and **Part B**. Part A contain 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer *all* questions. Part B contains 2 questions from each module of which student should answer *any one*. Each question can have a maximum 2 subdivisions and carry 6 marks.

Sample Course Level Assessment Questions

Course Outcome 1 (CO 1):

1. Explain the concept of DNA
2. Explain the concept of RNA.
3. Illustrate the concept of translation and transcription.
4. Discuss Gnome project and its impact on bioinformatics

Course Outcome 2 (CO 2):

1. Explain the features of biological databases?
2. Discuss primary sequence databases and secondary sequence databases.
3. Classify the two important classification schemes of structure classification databases.
4. Retrieve the sequence from primary / secondary databases.
5. Use of BLAST for comparing sequences.

Course Outcome 3 (CO 3):

1. Explain the importance of scoring matrices in sequence alignment.
2. Explain the different algorithms used for sequence alignment .
3. Illustrate Local and global alignment Algorithm for the sequence CGTGAATTCAT (sequence#1 or A) GACTTAC (sequence #2 or B)
4. Compute the best alignment of these two sequences: ACTGATTCA ACGCATCA Using -2 as a gap penalty, -3 as a mismatch penalty, and 2 as the score for a match.

Course Outcome 4 (CO 4):

1. Explain the Prokaryotic gene structure
2. Explain the Eukaryotic gene structure
3. Demonstrate the usage of Open Reading Frame with an example
   a. 5′-ATCTAAAATGGGTGCC-3′
4. Explain the working principle of microarray

Course Outcome 5 (CO 5):

1. Differentiate between the different protein molecular structure visualizations.
2. Use Web-based Map Viewer program, RasMol, PyMol data visualization techniques in bioinformatics .
3. Use PubMed to search for a particular pattern to specify the importance of mining the biomedical literature for data on functions to complement the sequence and structure data mined from nucleotide and protein databases.
4. Compare any three machine learning technologies and their applicability to data mining methods.

Model Question Paper

Course Code: S20MCA287

Course Name: BIOINFORMATICS

Max.Marks :60 Duration: 3 Hrs

Part A

Answer all questions. Each question carries 3 marks (10 * 3 = 30 Marks)

1. Write short note on “how genome carries hereditary data from organisms.”
2. What is antisense RNA?
3. Write a short note on primary database.

4. Write note on : (a) SCOP (b) CATH.
5. How many times faster is to find the best alignment for the sequences "RQQEPURSTC"
   and "QQESGPVRST" using N_W algorithm compared to assessing each possible alignment one by one?
6. Define raw score, bit score and e-value in BLAST.
7. Write a short note on of process is gene expression in Prokaryotic.
8. Justify the reasons for the high Prokaryotic gene density compared to Eukaryotes.
9. Differentiate between centralized and distributed data mining Infrastructure. Use diagrams if necessary.
10. Explain the significance of Hidden Markov Model in bioinformatics. Draw a sample Markov chain which is the basis of HMM, also mention how HMM is different from Markov chain.

Part B

Answer all questions. Each question carries 6 marks. (5 * 6 = 30 Marks)

11 What is the Central dogma of Molecular biology? How can Molecular biology be considered as an information science? 6

OR

12 With a neat diagram describe the structural and functional differences between DNA and RNA? 6

13 Explain different types of protein databases and its applications in bioinformatics. 6

OR

14 Differentiate between Composite protein sequence database and secondary databases . 6

15 Align the following sequence using Needleman and Wunch algorithm for global alignment ATTGC and AGGC with match +1, mismatch -1 and gap penalty -2. What is the score of the optimal global alignment? 6

OR

16 Find the best local alignment between ACCTAGG and GGCTCAATCA with +2 for a match, -1 for a mismatch and -2 for a gap using Smith Waterman Algorithm. 6

Explain Prokaryotic Gene structure with neat diagram.

17 What is GC content? How it differs in eukaryotic and prokaryotic genomes. 6

OR

18 Describe with the help of a diagram the generation of cDNAs. Mention its use and also write notes on ESTs. 6

19 Illustrate with the help of a neat diagram the pattern recognition and the label discovery process. 6

OR

20 Justify the importance of user interfaces in data visualization. With the support of a representative block diagram explain the structure of a 3D protein visualization tool. Also explain the UI components of the same. 6

Syllabus

Module 1: Computational Biology and Bioinformatics (7 Hours)
Computational Biology: Cell - Central Dogma of Molecular Biology - Structure of DNA, RNA and Protein - Coding and Non-coding RNAs - mRNA, tRNA, miRNA and siRNA. Bioinformatics: Nature & Scope of Bioinformatics, Gnome projects, Importance of bioinformatics, Pattern recognition and prediction.
Module 2: Biological Databases (8 Hours)
Biological Databases, Primary Sequence Databases, Composite protein sequence databases, Secondary Databases, Composite protein pattern databases, Structure classification databases. //Tutorial class may be arranged to the introduction and use of sequence retrieval from the databases.
Module 3: Data Searches and pairwise Alignment (10 Hours)
Dot Plots, Concept of Simple Alignment, Scoring matrices: Introduction to PAM & Blosum, Needleman and Wunsch Algorithm, Global and Local Alignments, Smith Waterman Algorithm, Multiple Sequence Alignment. Familiarize Database search tools: BLAST & FastA //Tutorial class may be arranged to the introduction and use of sequence alignment and BLAST.
Module 4: Genomics and Gene recognition (10 Hours)
Introduction to Gene expression in prokaryotes, Prokaryotic Gene structure, GC content in prokaryotic genomes, Gene Density. Eukaryotic Genomes: Gene structure, GC content in eukaryotic genomes, Gene Expression – Introduction to Microarrays.
Module 5: Data Visualization, Data mining and Machine learning (10 Hours)
Data Visualization - Introduction, Sequence Visualization, Structure Visualization, User Interface, Animation Versus Simulation, General-Purpose Technologies.

Data Mining using biological data, Methods, Infrastructure, Pattern recognition and discovery, Genetic Algorithms, Neural networks using biological data, Statistical methods using biological data, Introduction to Hidden Markov Models and Text mining.

//Tutorial class may be arranged to introduce and use - RasMol and PyMol .

Text Books

1. Dan. E. Krane and M. L. Raymer, "Fundamental Concepts of Bioinformatics", Pearson Education, 2003(Module 4)
2. Bryan Bergeron, M.D, "Bioinformatics Computing", Pearson Education, 2015. (Module 1,5)
3. Attwood T. K. and D. J. Parry-Smith," Introduction to Bioinformatics", Pearson Education, 2003 (Module 2,3)
4. Neil C Jones and Pavel A Pevzner, "An Introduction to Bioinformatics Algorithms", MIT Press, 2004

Reference Books

1. Jean-Michel Claverie and Cedric Notredame, "Bioinformatics For Dummies" , 2nd Edition,Wiley Publishing
2. David W Mount, "Bioinformatics- Sequence and Genome Analysis " , 2/e, Cold Spring Harbor
3. Laboratory Press, New York.
4. "Bioinformatics for Dummies" J. Claverie & C. Notredame ,Wiley India..

Web Reference

1. https://nptel.ac.in/courses/102/106/102106065/

Course Contents and Lecture Schedule

No	Topic	No. of Lectures
1	**Computational Biology and Bioinformatics**	**(7 Hours)**
1.1	Cell - Central Dogma of Molecular Biology, Structure of DNA	1
1.2	RNA and Protein: Coding and Non-coding RNAs -mRNA	1
1.3	tRNA, miRNA and siRNA	1
1.4	Nature & Scope of Bioinformatics, Gnome projects	1
1.5	Importance of bioinformatics, Pattern recognition and prediction	1
1.6	Folding problem	1
1.7	Sequence analysis, homology and analogy	1

2	**Biological Databases**	**(8 Hours)**
2.1	Primary Sequence databases: Nucleic acid and Protein sequence: PIR, MIPS, SWIS-PROT	1
2.2	Protein sequence: TrEMBL, NRL-3D	1
2.3	Composite protein sequence Databases: NRDB, OWL, MIPSX and SWISS-PROT+TrEMBL	1
2.4	Secondary Databases, Need for Secondary databases	1
2.5	Prosite	1
2.6	Prints	1
2.7	Blocks, Profile, Pfam, Identify	1
2.8	Composite Protein Pattern Database and Structure Classification Databases	1
3	**Data Searches and pairwise Alignment**	**(10 Hours)**
3.1	Dot Plots	1
3.2	Concept of Simple Alignment, GAPS	1
3.3	Scoring matrices	1
3.4	Introduction to PAM	1
3.5	Introduction to Blosum	1
3.6	Needleman and Wunsch Algorithm	1
3.7	Global and Local Alignments: Semiglobal alignment	1
3.8	Smith Waterman Algorithm	1
3.9	Alignment scores and statistical significance of database search , Multiple Sequence Alignment.	1
3.10	Familiarize Database search tools: BLAST & FastA	1
4	**Gene structure and expression of Prokaryotic and Eukaryotes.**	**(10 Hours)**
4.1	Introduction to Gene expression in Prokaryotes	1
4.2	Prokaryotic Gene structure	1
4.3	GC content in prokaryotic genomes	1
4.4	Prokaryotic Genomes -Gene Density	1
4.5	Eukaryotic Genomes	1
4.6	Gene structure , ORF in Prokaryotic	1

4.7	GC content in Eukaryotic Genomes	1
4.8	Gene Expression - cDNAs & ESTs,	1
4.9	Serial Analysis of Gene Expression	1
4.10	Introduction to Microarrays.	1
5	**Data Visualization, Data mining & Machine learning**	**(10 Hours)**
5.1	Data Visualization Introduction	1
5.2	Sequence Visualization- Sequence Map	1
5.3	Structure Visualization- Rendering tools	1
5.4	User Interface - User Interface Components, Alternative Metaphors, Display Architecture	1
5.5	Animation Versus Simulation, General-Purpose Technologies.	1
5.6	Data Mining, Methods, Infrastructure	1
5.7	Pattern recognition and discovery	1
5.8	Genetic Algorithms	1
5.9	Neural networks, Statistical methods	1
5.10	Hidden Markov Models and Text mining	1

20MCA289	**SOCIAL NETWORK ANALYSIS**	**CATEGORY**	**L**	**T**	**P**	**CREDIT**
		ELECTIVE	**3**	**1**	**0**	**4**

Preamble: This course intends to provide insight into social network analysis. The objective of this course is to enable students analyse and visualize network data. This course will create an understanding about the semantic web, structure of various social networks and the structure of search engines.

Prerequisite: Basic concepts of graph theory and networks

Course Outcomes: After the completion of the course the student will be able to

CO No.	Course Outcome (CO)	Bloom's Category Level
CO 1	Explain the basic concepts of semantic web and social network analysis.	Level 2: Understand
CO 2	Describe the ontology-based knowledge representation techniques in social network.	Level 2: Understand
CO 3	Discuss aggregation of social network information and representation of social individuals and social relationships.	Level 2: Understand
CO 4	Describe the structure of the Web and Facebook as a graph and the algorithms for searching and community discovery.	Level 2: Understand
CO 5	Explain the general architecture of a search engine and specifically the Google search engine architecture.	Level 2: Understand

Mapping of course outcomes with program outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	2	2					1					
CO 2	2	2					1					
CO 3	2	2					2					
CO 4	2	3		2	2	2	2			2		
CO 5	2	3		2	2		2					

3/2/1: High/Medium/Low

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination
	1	2	
Remember	15	15	20

Understand	35	35	40
Apply			
Analyze			
Evaluate			
Create			

Mark distribution

Total Marks	CIE	ESE	ESE Duration
100	40	60	3 hours

Continuous Internal Evaluation Pattern:

Attendance	: 8 marks
Continuous Assessment Test (2 numbers)	: 20 marks
Assignment/Quiz/Course project	: 12 marks

End Semester Examination Pattern: There will be two parts; Part A and Part B. Part A contains 10 compulsory short answer questions, 2 from each module. Each question carries 3 marks. Part B contains 2 questions from each module of which student should answer any one. Each question can have maximum 2 sub-divisions and carry 6 marks.

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Explain the development of semantic Web and the emergence of Social Web.
2. Describe the global structure social networks.
3. Discuss in detail about the macro-structure of social networks.
4. “Most network analysis methods work on an abstract, graph-based representation of real-world networks”. Justify this statement.

Course Outcome 2 (CO2)

1. Describe the characteristics of Resource Description Framework (RDF).
2. Compare the features of Web Ontology Language (WOL) and Unified Modeling Language (UML).
3. Compare the features of Web Ontology Language (WOL) and Entity Relationship (ER) Model.

Course Outcome 3(CO3):

1. Describe the ontological representation of social individuals.
2. Explain the generic architecture of Semantic Web applications.
3. Discuss how semantic web applications can be built with social network features?

Course Outcome 4 (CO4):

1. Describe Zipf's Law.
2. Write the limitations of HyperANF Algorithm and explain how it can be sorted out using the Iterative Fringe Upper Bound (iFUB) Algorithm.
3. What is meant by Degree Assortativity? What is the use of this measure?
4. "A user who logs in more generally has more friends on Facebook", describe how can we conclude this statement.

Course Outcome 5 (CO5):

1. Draw the architecture of a general search engine and explain how it works.
2. Explain how the HITS Algorithm works to assign ranks to web pages.
3. Compare the HITS Algorithm and the Page Rank Algorithm.

Model Question Paper
Course Code: 20MCA289

Course Name: SOCIAL NETWORK ANALYSIS

Max. Marks :60 **Duration: 3 Hrs**

Part A

Answer all questions. Each question carries 3 marks (10 * 3 = 30 Marks)

1. What is meant by semantic web?
2. Write notes on personal networks.
3. Define Electronic discussion networks.
4. List out the features of blogs that can be used for social network extraction.
5. Explain how the reasoning with instance equality is done in social network data?
6. What is meant by Evaluating Smushing?
7. Define "Power Law".
8. What is "Spid"? How it is used to differentiate between web-network and social network?
9. What are the basic functions of the storage repository of a search engine?
10. How can we identify web spam pages?

Part B

Answer all questions. Each question carries 6 marks. (5 * 6 = 30 Marks)

11. List and explain various measures in network analysis. 6 marks

OR

12. Describe the macro-structure of social networks. 6 marks

13. What is meant by ontology-based knowledge representation? Explain its role in the semantic web. 6 marks

OR

14. Compare the features of Web Ontology Language (WOL) and Extensible Markup Language (XML). 6 marks

15. Describe how aggregating and reasoning can be done on social network data. 6 marks

OR

16. Discuss the ontological representation of social relationships. 6 marks

17. Define the following with suitable example:
    a) Rank exponent 2 marks
    b) Hop plot exponent 2 marks
    c) Eigen exponent 2 marks

OR

18. Explain how to generate in-degree and out-degree distributions on the graph of the Web crawl. 6 marks

19. Describe how the web crawler module in a search engine does the page selection and page refresh. 6 marks

OR

20. Draw the architecture of Google search engine and comment on each of its components. 6 marks

Syllabus

Module I (9 Hours)
Introduction to the Semantic Web and Social Networks: The Semantic Web, Limitations of the current Web, The semantic solution, Development of the Semantic Web, The emergence of the social web, Social Network Analysis, Development of Social Network Analysis, The global structure of networks, The macro-structure of social networks, Personal networks.
Module II (8 Hours)
Electronic sources for network analysis: Electronic discussion networks, Blogs and online communities, Web-based networks.

Knowledge Representation on the Semantic Web: Ontologies and their role in the Semantic Web, Ontology languages for the Semantic Web, The Resource Description Framework (RDF) and RDF Schema, The Web Ontology Language (OWL), Comparison of Ontology languages with the Unified Modelling Language (UML), Comparison to the Entity/Relationship (E/R) model and the Relational model, Comparison to the Extensible Markup Language (XML) and XML Schema.
Module III (8 Hours)
Modelling and aggregating social network data: Network data representation, Ontological representation of social individuals, Ontological representation of social relationships, Aggregating and reasoning with social network data, Representing identity, On the notion of equality, Determining equality, Reasoning with instance equality, Evaluating smushing.
Module IV (10 Hours)
Graph Structure of the Web: Breadth First Search (BFS) Algorithm, Strongly Connected Components (SCC) Algorithm, Weakly Connected Components (WCC) Algorithm, In-degree and out- degree distributions, Connected Components, Zipf’s Law, Rank Exponent R, Out-Degree Exponent O, Hop Plot Exponent H, Eigen Exponent E. **Graph Structure of Facebook:** Hyper ANF Algorithm, Iterative Fringe Upper Bound (iFUB) Algorithm, Spid, Degree Distribution, Path Length, Component Size, Clustering Coefficient and Degeneracy, Friends-of-Friends, Degree Assortativity, Login Correlation, Effects of Age, Gender and Country of Origin.
Module V (10 Hours)
Link Analysis: Search Engine – Search engine architecture, Crawling, Storage, Indexing, Ranking, HITS Algorithm, Page rank algorithm, Random walk, SALSA Algorithm, Bayesian Algorithm; Google - Google architecture, Data Structures, Crawling, Searching, Web Spam Pages.

Textbooks.

1. Social Networks and the Semantic Web, Peter Mika, Springer, 2007. (For Modules 1, 2 & 3)
2. Practical Social Network Analysis with Python, Krishna Raj P. M., Ankith Mohan, K. G. Srinivasa, Springer, 2018. (For Modules 4 & 5)

References

1. Social Network Analysis, John Scott, SAGE Publications, 4th Edition (2017)
2. Social Network Analysis - Interdisciplinary Approaches and Case Studies, Xiaoming Fu, Jar-Der Luo and Margarete Boos, CRC Press (2017)
3. Handbook of Social Network Analysis, John Scott and Peter J. Carrington, SAGE Publications (2011)

4. Social Network Analysis - Methods and Applications, Stanley Wasserman and Katherine Faust, Cambridge University Press (2012)

Web - References

1. https://onlinecourses.nptel.ac.in/noc20_cs78/preview
2. https://www.coursera.org/learn/social-network-analysis
3. https://www.coursera.org/learn/python-social-network-analysis

Course Contents and Lecture Schedule

No	Topic	No. of Lectures
1	**Module 1**	**9 Hours**
1.1	The Semantic Web, Limitations of the current Web	1
1.2	The semantic solution	1
1.3	Development of the Semantic Web	1
1.4	The emergence of the Social Web	1
1.5	Social Network Analysis	1
1.6	Development of Social Network Analysis	1
1.7	The global structure of networks	1
1.8	The macro-structure of social networks	1
1.9	Personal networks	1
2	**Module 2**	**8 Hours**
2.1	Electronic sources for network analysis, Electronic discussion networks	1
2.2	Blogs and online communities	1
2.3	Web-based networks	1
2.4	Knowledge Representation on the Semantic Web	1
2.5	Ontologies and their role in the Semantic Web	1
2.6	Ontology languages for the Semantic Web, The Resource Description Framework (RDF) and RDF Schema	1
2.7	The Web Ontology Language (OWL), Comparison of Ontology languages with the Unified Modelling Language (UML)	1
2.8	Comparison to the Entity/Relationship (E/R) model and the Relational model, Comparison to the Extensible Markup Language (XML) and XML Schema	1
3	**Module 3**	**8 Hours**
3.1	Modelling and aggregating social network data, Network data representation	1
3.2	Ontological representation of social individuals	1
3.3	Ontological representation of social relationships	1
3.4	Aggregating and reasoning with social network data	1
3.5	Representing identity	1
3.6	Notion of equality, Determining equality	1

3.7	Reasoning with instance equality	1
3.8	Evaluating smushing	1
4	**Module 4**	**10 Hours**
4.1	Graph Structure of the Web	1
4.2	Breadth First Search (BFS) Algorithm	1
4.3	Strongly Connected Components (SCC) Algorithm, Weakly Connected Components (WCC) Algorithm	1
4.4	In-degree and out- degree distributions, Connected Components	1
4.5	Zipf’s Law	1
4.6	Rank Exponent R, Out-Degree Exponent O, Hop Plot Exponent H, Eigen Exponent E	1
4.7	Graph Structure of Facebook: HyperANF Algorithm	1
4.8	Iterative Fringe Upper Bound (iFUB) Algorithm, Spid, Degree Distribution, Path Length	1
4.9	Component Size, Clustering Coefficient and Degeneracy, Friends-of-Friends	1
4.10	Degree Assortativity, Login Correlation, Effects of Age, Gender and Country of Origin	1
5	**Module 5**	**10 Hours**
5.1	Link Analysis: Search Engine – Search engine architecture	1
5.2	Crawling, Storage, Indexing	1
5.3	Ranking, HITS Algorithm	1
5.4	Page rank algorithm	1
5.5	Random walk	1
5.6	SALSA Algorithm	1
5.7	Bayesian Algorithm	1
5.8	Google - Google architecture	1
5.9	Data Structures, Crawling, Searching	1
5.10	Web Spam Pages	1

20MCA241	DATA SCIENCE LAB	CATEGORY	L	T	P	CREDIT
		LAB	0	1	3	2

Preamble: This is an introductory practical course on Data Science and student will learn how to use various scientific libraries in python to implement data mining techniques and machine learning algorithms.

Prerequisite: Fundamentals of programming, python programming fundamentals, Machine learning, fundamentals of web programming,

Course Outcomes: After the completion of the course the student will be able to

CO No.	Course Outcome (CO)	Bloom's Category Level
CO 1	Use different python packages to perform numerical calculations, statistical computations and data visualization	Level 3: Apply
CO 2	Use different packages and frameworks to implement regression and classification algorithms.	Level 3: Apply
CO 3	Use different packages and frameworks to implement text classification using SVM and clustering using k-means	Level 3: Apply
CO 4	Implement convolutional neural network algorithm using Keras framework.	Level 3: Apply
CO 5	Implement programs for web data mining and natural language processing using NLTK	Level 3: Apply

Mapping of course outcomes with program outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	3	3	3	1	3	2	3		2			
CO 2	3	3	3	2	3	2	3		2			
CO 3	3	3	3	2	3	2	3		2			
CO 4	3	3	3	2	3	2	3		2			
CO 5	3	3	3	2	3	3	3		2			

3/2/1: High/Medium/Low

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination
	1	2	
Remember (K1)			
Understand (K2)			
Apply (K3)	50	50	50
Analyse (K4)			
Evaluate (K5)			
Create(K6)			

Mark distribution

Total Marks	CIE	ESE	ESE Duration
100	50	50	3 hours

Continuous Internal Evaluation Pattern:

Maximum Marks: 50	
Attendance	7½
Maintenance of daily lab record and GitHub management	10
Regular class viva voce	7½
Timely completion of day-to-day tasks	10
Tests/Evaluation	15

End Semester Examination Pattern:

<table>
<tr><th colspan="4">Maximum Marks: 50</th></tr>
<tr><td colspan="3">Verification of Daily program record and Git Repository</td><td>5 marks</td></tr>
<tr><td colspan="3">Viva</td><td>10 marks</td></tr>
<tr><td rowspan="4">Problem solving (Based on difficulty level, one or more questions may be given)</td><td>Flowchart / Algorithm / Structured description of problem to explain how the problem can be solved / Interface Design</td><td>15%</td><td rowspan="4">35 marks</td></tr>
<tr><td>Program correctness</td><td>50%</td></tr>
<tr><td>Code efficiency</td><td>15%</td></tr>
<tr><td>Formatted output</td><td>20%</td></tr>
</table>

Course Level Assessment Questions

Course Outcome 1 (CO1):

- Review of python programming – Programs review the fundamentals of python (simple python programs ice breaker) – (at most one lab session)

- Matrix operations (using vectorization) and transformation using python and SVD using Python.
- Programs using matplotlib / plotly / bokeh / seaborn for data visualisation.
- Programs to handle data using pandas.

Course Outcome 2 (CO2)

- Program to implement k-NN classification using any standard dataset available in the public domain and find the accuracy of the algorithm.
- Program to implement Naïve Bayes Algorithm using any standard dataset available in the public domain and find the accuracy of the algorithm
- Program to implement linear and multiple regression techniques using any standard dataset available in the public domain and evaluate its performance.

Course Outcome 3(CO3):

- Program to implement text classification using Support vector machine.
- Program to implement decision trees using any standard dataset available in the public domain and find the accuracy of the algorithm.
- Program to implement k-means clustering technique using any standard dataset available in the public domain

Course Outcome 4 (CO4):

- Programs on feedforward network to classify any standard dataset available in the public domain.
- Programs on convolutional neural network to classify images from any standard dataset in the public domain.

*[Note] : Encourage students to refer standard neural network architectures such as LeNet5, ResNet, GoogLeNet etc. and use these as starting points for their models.

Course Outcome 5 (CO5):

<u>Web Data Mining</u>

- Implement a simple web crawler (ensure ethical conduct).
- Implement a program to scrap the web page of any popular website – suggested python package is scrappy (ensure ethical conduct).

Natural Language Processing

Problems may be designed for the following topics so that students can get hands on experience in using python for natural language processing:

- Part of Speech tagging
- N-gram and smoothening
- Chunking

Syllabus
Review of python programming, Matrix operations, Data Visualisation using matplotlib / plotly / bokeh / seaborn, Data handling using pandas, Classification k-NN algorithm, Naïve Bayes algorithm, Implementation of linear and multiple regression techniques, Text classification using Support vector machine, Implementation of Decision Trees, Clustering using k-means algorithm, Convolutional Neural Network to classify images using Keras framework, Web Crawler and Scrapping web pages, Implementation of NLP - Part of Speech tagging, N-gram & smoothening and Chunking using NLTK.

Reference Books

1. Christopher M Bishop, “Pattern Learning and Machine Learning”, Springer, 2006
2. E. Alpayidin, “Introduction to Machine Learning”, Prentice Hall of India (2005)
3. T. Hastie, RT Ibrashiran and J. Friedman, “The Elements of Statistical Learning”, Springer 2001
4. Toby Segaran, “Programming Collective Intelligence: Building Smart Web 2.0 Applications”, O&#39; Reilly Media; 1 edition (16 August 2007).
5. Drew Conway, John Myles White, “Machine Learning for Hackers: Case Studies and
6. Algorithms to Get You Started”, O&#39; Reilly Media; 1 edition (13 February 2012)
7. Simon Rogers, Mark Girolami, “A First course in Machine Learning”, CRC Press, First Indian reprint, 2015.
8. Tom Mitchell, “Machine Learning”, McGraw Hill, 1997.
9. Bing Liu, Web Data Mining - Exploring Hyperlinks, Contents and Usage Data, Second edition, Springer 2011

Course Contents and Lab Schedule

Sl No.	Topic	No. of hours
1	Review of python programming, Matrix operations, Programs using matplotlib / plotly / bokeh / seaborn for data visualisation and programs to handle data using pandas.	8
2	Program to implement k-NN classification using any standard dataset available in the public domain and find the accuracy of the algorithm	2
3	Program to implement Naïve Bayes Algorithm using any standard dataset available in the public domain and find the accuracy of the algorithm	2
4	Program to implement linear and multiple regression techniques using any standard dataset available in the public domain and evaluate its performance.	4
5	Program to implement text classification using Support vector machine.	4
6	Program to implement decision trees using any standard dataset available in the public domain and find the accuracy of the algorithm	4
7	Program to implement k-means clustering technique using any standard dataset available in the public domain	2
8	Program on convolutional neural network to classify images from any standard dataset in the public domain using Keras framework.	6
9	Program to implement a simple web crawler and scrapping web pages.	6
10	Implement problems on natural language processing - Part of Speech tagging, N-gram & smoothening and Chunking using NLTK	8

20MCA243	MOBILE APPLICATION DEVELOPMENT LAB	CATEGORY	L	T	P	CREDIT
		LAB	0	1	3	2

Preamble: This is a practical course on Mobile Application Development and student will learn how to program in Android Platform and develop applications using SQLite that run on Andriod Operating System.

Prerequisite: Basic knowledge on programming and database concepts.

Course Outcomes: After the completion of the course the student will be able to

CO No.	Course Outcome (CO)	Bloom's Category Level
CO 1	Design and develop user interfaces for mobile apps using basic building blocks, UI components and application structure using Emulator	Level 3: Apply
CO 2	Write simple programs and develop small applications using the concepts of UI design, layouts and preferences	Level 3: Apply
CO 3	Develop applications with multiple activities using intents, array adapter, exceptions and options menu.	Level 3: Apply
CO 4	Implement activities with dialogs, spinner, fragments and navigation drawer by applying themes	Level 3: Apply
CO 5	Develop mobile applications using SQLite.	Level 3: Apply

Mapping of course outcomes with program outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	3	3	3	1	3	2	3		2			
CO 2	3	3	3	2	3	2	3		2			
CO 3	3	3	3	2	3	2	3		2			
CO 4	3	3	3	2	3	2	3		2			
CO 5	3	3	3	2	3	3	3		2			

3/2/1: High/Medium/Low

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination
	1	2	
Remember(K1)			
Understand(K2)			
Apply(K3)	50	50	50
Analyse(K4)			
Evaluate(K5)			
Create(K6)			

Mark distribution

Total Marks	CIE	ESE	ESE Duration
100	50	50	3 hours

Continuous Internal Evaluation Pattern:

Maximum Marks: 50	
Attendance	7½
Maintenance of daily lab record and GitHub management	10
Regular class viva voce	7½
Timely completion of day-to-day tasks	10
Tests/Evaluation	15

End Semester Examination Pattern:

Maximum Marks: 50			
Verification of Daily program record and Git Repository			5 marks
Viva			10 marks
Problem solving (Based on difficulty level, one or more questions may be given)	Flowchart / Algorithm / Structured description of problem to explain how the problem can be solved / Interface Design	15%	35 marks
	Program correctness	50%	
	Code efficiency	15%	
	Formatted output	20%	

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Design a Login Form with username and password using LinearLayout and toast valid credentials
2. Write a program that demonstrates Activity Lifecycle.
3. Implementing basic arithmetic operations of a simple calculator
4. Implement validations on various UI controls

Course Outcome 2 (CO2)

1. Design a registration activity and store registration details in local memory of phone using Intents and SharedPreferences
2. Design a simple Calculator using GridLayout and Cascaded LinearLayout
3. Create a Facebook page using RelativeLayout; set properties using .xml file
4. Develop an application that toggles image using FrameLayout

Course Outcome 3(CO3):

1. Implement Adapters and perform exception handling
2. Implement Intent to navigate between multiple activities
3. Develop application that works with explicit intents
4. Implement Options Menu to navigate to activities
5. Develop an application that uses ArrayAdapter with ListView.

Course Outcome 4 (CO4):

1. Develop an application that use GridView with images and display Alert box on selection
2. Develop an application that implements Spinner component and perform event handling
3. Apply themes via code and manifest file
4. Develop application using Fragments
5. Implement Navigation drawer

Course Outcome 5 (CO5):

1. Create database using SQLite and perform INSERT and SELECT
2. Perform UPDATE and DELETE on SQLite database
3. Develop an application as a micro project which uses SQLite database as an assignment

Syllabus

Fundamentals: Basic Building blocks – Activities, Services, Broadcast Receivers and Content providers, UI Components – Views and notifications Components for communication -Intents and Intent Filters
Application Structure: AndroidManifest.xml, user-permission – sdk, Resources and R.java, Assets, Layouts and Drawable Resources, Activities and Activity lifecycle.
Emulator-Android Virtual Device: Launching emulator, Editing emulator settings, Emulator shortcuts, Logcat usage, Introduction to DDMS
Basic UI design: Form widgets, Text Fields, Validation of EditText, Layouts, [dip, dp, sip, sp] versus px
Preferences: Shared Preferences, Preferences from xml
Menu: Option menu, Context menu, menu from xml, menu via code
Intents: Explicit Intents, Implicit intents
UI design: Time and Date, Images and media, Android Adapter and ListView, Composite, Alert Dialogs and Toast, Popup, Fragments, Navigation drawer
Tabs, Tab Activity Styles & Themes: styles.xml, drawable resources for shapes, gradients (selectors), style attribute in layout file, Applying themes via code and manifest file
Content Providers: SQLite Programming, SQLite Open Helper, SQLite Database, Cursor, Reading and updating Contacts, Reading bookmarks

Reference Books

1. Joseph Annuzzi Jr, Lauren Darcey, Shane Condor, “Advanced Android Application Development, Developers Library”, Pearson Education, 4th Edition (2015)

2. Lauren Darcey, Shane Condor, “Android, Wireless Application Development”, Pearson Education, 3rd Edition.

3. Paul Deitel, Harvey Deitel, Alexander Wald, “Android 6 for programmers, An AppDriven Approach”, Pearson Education

4. Rap Payne, “Beginning App Development with Flutter: Create Cross-Platform Mobile Apps”, Apress (2019)

Course Contents and Lecture Schedule

Sl No	Topic	No. of hours
1	Fundamentals – Basic building blocks	3
2	Application structure, layout and resources	3
3	Android Virtual Device, Activity Lifecycle	3
4	Basic UI Design and EditText Validation	4
5	Shared Preferences, RelativeLayout, FrameLayout, GridLayout and Preferences from xml	9
6	ArrayAdapter, ListView and Exception handling	3
7	Various Menu options	3
8	Explicit and Implicit Intents	3
9	Images and media, Dialogs, Spinner component, Popups, Fragments, Navigation drawer	6
10	Applying themes and styles .xml	3
11	SQLite Programming	6

CODE	COURSE NAME	CATEGORY	L	T	P	CREDIT
20MCA245	MINI PROJECT	PROJECT	-	-	4	2

Preamble: This project work aims to enable the students to apply the software engineering principles on a real software project, to make the students familiar with the stages of a deployment pipeline and to develop a software product using the latest software development methodology.

Prerequisite: Knowledge in software engineering principles and programming skills.

Course Outcomes: After the completion of the course the student will be able to

CO No.	Course Outcome (CO)	Bloom's Category Level
CO 1	Identify a real-life project which is useful to society / industry	Level 2: Understand
CO 2	Interact with people to identify the project requirements	Level 3: Apply
CO 3	Apply suitable development methodology for the development of the product / project	Level 3: Apply
CO 4	Analyse and design a software product / project	Level 4: Analyse
CO 5	Test the modules at various stages of project development	Level 5: Evaluate
CO 6	Build and integrate different software modules	Level 6: Create
CO 7	Document and deploy the product / project	Level 3: Apply

Mapping of course outcomes with program outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	2	3	3	3	1	2	3	3	3	3	3	3
CO 2	2	3	2	3	2	3	2	1	3	2	3	
CO 3	3	3	3	3	3	1	3	3	1		2	
CO 4	3	3	3	3	3		3	3	1	1	2	
CO 5	3	3	3	3	3		2	3			1	
CO 6	3	3	3	3	3	2	3	3		2	3	3
CO 7	1	1	3	3	3	2	3	3	2	1	2	

3/2/1: High/Medium/Low

Mark distribution

Total Marks	CIE	ESE
100	100	-

Assessment Criteria

Class participation and attendance	10%
Evaluation	50%
Class work	40%

Marks Division

Continuous evaluation by Supervisor, Scrum Master and Project Guide	50 Marks
Interim evaluation by the Project Assessment Board	25 Marks
Final evaluation by the Project Assessment Board	25 Marks
Total	**100 Marks**

Guidelines:

- Students shall identify Real-Life Projects which are relevant and useful to the society or industry.
- The project shall be an individual project and must be done in-house. The student has to spend time in the lab for the project work.
- Attendance as per MCA regulations is applicable for submitting the project for final evaluation.
- Students shall submit project synopsis and get prior approval from the Project (Faculty) Supervisor before the project work begins.
- If there is a customer for the project then he/she will be the Product Owner (External Guide) and a faculty from the department will be the Internal Guide. If there is no such customer then the Internal Guide himself/herself shall act as the Product Owner.
- A faculty / technical staff shall act as the Scrum Master to continuously monitor the project development. Periodic meetings, of less than 15 minutes, at the convenience of

the Scrum Master are to be highly encouraged. Ensure such meetings occur once in three days.

- Set a sprint as two weeks, ensure biweekly reviews. A review shall not exceed 30 minutes. A demo to the Product Owner (Project Guide) is mandatory in every review.

- The student shall maintain a Scrum Book (Rough Record) which has to be divided into 4 parts – (i) Product Backlog (ii) Database & UI Design (iii) Testing & Validation and (iv) Details of Versions. Make dated entries in the corresponding part at regular intervals. The corrections and comments from Product Owner and Scrum Master should be clearly indicated with the Date.

- Test Driven Development methodology may be practiced for the project development. BugZilla, BackLog or any such tool may be used for Bug Tracking.

- Git shall be used for Version Control and Git commit history may be verified as part of project evaluation .

- LaTeX or an equivalent tool shall be used for preparing Presentations and Project Report.

- Interim evaluations of project's progress shall be conducted as part of Internal Assessment. Project Evaluation Board may consist of Project Supervisor, Product Owner, Scrum Master and one other Faculty Member from the department. Scrum reviews shall not be sacrificed for such presentations.

- At the end of the semester entire project development activities shall be evaluated internally by the Project Evaluation Board.

Week	**Schedule**
1	Familiarisation with build tools (editor/IDE, compiler such as gcc with commonly used options/switches, debugger like dgb). Familiarisation with an IDE (Eclipse, NetBeans…), that supports build tools and common version control operations using Git . Familiarisation with Docker Selection of Topic, Formation of Development Team, Feasibility analysis.
2	Topic Approval, Meeting of Development Team including Scrum Master with Product Owner. Informal, preliminary discussions of requirements. Creating user stories in the rough record.

	Commencement of the Project.
3	Identifying modules, Initial Design of Database & UI. Creating a Docker container for the environment Creating an empty git repository by Scrum Master / one member of the Development team and setting permission to other members. Pushing the first version of the Project along with a Readme file containing contact details of team members. Creating pull requests for sample update of Readme by each member and merging the pull requests of one by another.
4-5	Setting up systems for development, testing and production. Design of the basic model of a simple deployment pipeline Creating a suitable folder structure (Maven's folder structure is desirable). Creating Unit tests using an XUnit framework, Writing the build and code analysis script, Writing acceptance test scripts and test cases, Setting up a Continuous Integration System like Jenkins. Automating acceptance tests with Selenium, Karate or an equivalent tool, writing a simple deployment script that uses scp/rsync or Ansible for copying the Dockerfile and running Docker with ssh. First Scrum Review. (Here onwards, the Scrum reviews are conducted on every other week)
7	Project Presentation - Interim Evaluation to be based on Git History
14	Submission of Project Report, with Scrum Book Project Presentation – Final Evaluation to be based on Git History, Scrum Book, Project Report and Presentation

References

1. Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation (Addison-Wesley Signature Series (Fowler)) 1st Edition
2. Alistair Cockburn, Agile Software Development: The Cooperative Game, Addison Wesley, 2nd Edition (2006).
3. Andrew Hunt, David Thomas, The Pragmatic Programmer: From Journeyman to Master, Pearson India, 1st Edition (2008).

4. Ken Schwaber, Mike Beedle, Agile Software Development with Scrum, Pearson (2008).
5. Lisa Crispin, Janet Gregory, Agile Testing: A Practical Guide for Testers and Agile Teams, Addison Wesley Professional, 1st Edition (2008).
6. Mike Cohn, User Stories Applied: For Agile Software Development, Addison Wesley, 1st Edition, (2004).
7. Pressman, R.S., Software Engineering: A Practitioner's Approach, McGraw Hill SE, 7th Edition, (2010).
8. Robert C. Martin, Agile Software Development, Principles, Patterns and Practices, Prentice Hall Imprint, Pearson Education, 2nd Edition (2002).\
9. Rod Stephens, Beginning Software Engineering, Wrox Series, Wiley India Pvt Ltd (2015).
10. RyPress Ry's Git Tutorial (Free e-book)

Web Reference

1. Introduction to DevOps (https://www.edx.org/course/introduction-devops-microsoft-dev212x)

CODE	COURSE NAME	CATEGORY	L	T	P	CREDIT
20MCANC3	Domain Expertise Workshops	Non-Credit Course	-	-	1	Nil

Preamble: This course intends to give insight into various application domains and technology domains in IT industry.

Prerequisite: Nil

Course Outcomes: After the completion of the course the student will be able to

CO No.	Course Outcome (CO)	Bloom's Category Level
CO 1	Associate real-life problems with IT solutions	Level 2: Understand
CO 2	Describe latest developments in IT field	Level 2: Understand
CO 3	Interact with technical experts	Level 3: Apply
CO 4	Prepare technical documents	Level 3: Apply
CO 5	Present a topic before an audience	Level 2: Understand

Mapping of course outcomes with program outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	1	2				3	3		3	3	3	3
CO 2	2	2				3	3		3	3	3	3
CO 3		2				3	3		3			3
CO 4						3			3			3
CO 5						3			3			3

3/2/1: High/Medium/Low

Mark distribution

Total Marks	CIE	ESE
-	-	-

Guidelines:

- As part of this course following activities shall be done:
    - ➢ Expert talks shall be arranged to explain about various Application domains like Retail, Finance, Healthcare, Automotive, Manufacturing… and IT domains like IoT, AI, Bigdata, Full Stack Development, Robotic Process Automation…
    - ➢ Instruct students to research and submit reports about any of these domains.
    - ➢ Instruct students to study about these domains and take seminars…
- One hour in every week or two hours in alternate weeks shall be used for this course
- Staff-in-charge shall maintain a file with the records, documents and reports as hardcopies or e-copies of all the activities done on this course.
- These documents shall be submitted for verification during the academic audit.

SEMESTER -4

CODE	COURSE NAME	CATEGORY	L	T	P	CREDIT
20MCA242	**COMPREHENSIVE VIVA**	**VIVA**	**-**	**-**	**-**	**6**

Preamble: Comprehensive Viva intends to assess the knowledge gained by a student in the core courses of this programme and to make the student aware of his/her knowledge level and where he/she stands after completing this programme. This course will help the student in preparing for comprehensive examinations and improve the confidence in answering questions in objective mode.

Prerequisite: Thorough knowledge in all the courses he/she learned during this programme.

Course Outcomes: After the completion of the course the student will be able to

CO No.	Course Outcome (CO)	Bloom's Category Level
CO 1	Articulate the concepts in the core courses learned through this programme.	Level 2: Understand
CO 2	Attend technical interviews with confidence.	Level 2: Understand
CO 3	Interpret questions and answer them with clarity.	Level 2: Understand
CO 4	Make use of the concepts learned through this programme in future.	Level 3: Apply

Mapping of course outcomes with program outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	3	3		2	2		2		3			
CO 2	3	3	1	2	3	2	3		3			
CO 3	1	2				2	2		3			
CO 4	3	2	3	2	2	3	3		2			

Mark distribution

Total Marks	CIE	ESE
100	-	100

Guidelines:

- Comprehensive viva shall be conducted within the first 20 days of the fourth semester.

- Viva shall be conducted by a panel of examiners consisting of:
  1. Head of the department
  2. A senior faculty in the department
  3. External examiner appointed by the university

- Viva shall be conducted for each student for a minimum of 20 minutes

- Knowledge level of the student shall be assessed on the following topics.
    - 20MCA105 - Advanced Data Structures
    - 20MCA107 - Advanced Software Engineering
    - 20MCA102 - Advanced Database Management Systems
    - 20MCA201 - Data Science & Machine Learning
    - 20MCA203 - Design & Analysis of Algorithms
    - Code snippets in Java or Python to solve simple problems.
    - Technologies used in the project work
    - Recent developments in the field of computer science.

CODE	COURSE NAME	CATEGORY	L	T	P	CREDIT
20MCA244	**SEMINAR**	**SEMINAR**	**-**	**-**	**2**	**2**

Preamble: This course intends to enable the students to gain knowledge in any of the technically relevant current topics on Computer Science or Information Technology, and to acquire confidence in presenting the topic and preparing a report.

Prerequisite: Nil

Course Outcomes: After the completion of the course the student will be able to

CO No.	Course Outcome (CO)	Bloom's Category Level
CO 1	Annotate the ideas presented in technical papers	Level 2: Understand
CO 2	Comprehend a concept by referring different technical documents	Level 2: Understand
CO 3	Prepare technical documents	Level 3: Apply
CO 4	Present a topic before an audience	Level 3: Apply
CO 5	Interact with the audience	Level 2: Understand

Mapping of course outcomes with program outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	2	3	1	3	2		3		3	2		2
CO 2	2	3	1	3	2		3		3	2		2
CO 3	2		1	2	3	2	3		3	2		2
CO 4	2	2			3	3			3		2	
CO 5	2	2			3	3			3		2	

Mark distribution

Total Marks	**CIE**	**ESE**
50	50	-

Assessment Criteria

Scope and relevance of topic	20%
Quality of presentation slides	10%
Presentation skills	30%
Knowledge in the topic	20%
Report	20%

Marks Division

Evaluation by Faculty Guide	20 Marks
Evaluation by the Faculty Committee	30 Marks
Total	**50 Marks**

Guidelines:

- Students shall conduct detailed study on a technically relevant current topic in Computer Science / Information Technology under the supervision of a Faculty Guide and present it as a seminar at the end of the study.
- The study may be conducted on
    - articles published in reputed journals/conference proceedings
    - recent development in Computer Science / Information Technology
    - recent research and development activity in a research lab
    - latest software tool or framework
- Students shall submit an abstract on identified topic and get prior approval from the Faculty Guide before the study begins.
- The student shall submit a seminar report, based on the study and their findings. The report shall not be a reproduction of original paper or manual.
- The study and its findings shall be presented in the class taking a duration of 15-20 minutes.
- LaTeX or an equivalent tool shall be used for preparing Presentations and Seminar Report.
- Students shall be encouraged to publish their study in journals and due credit shall be given to such students.
- A committee of three senior faculty members shall constituted by the head of the department and the seminar presentation shall be evaluated by that committee.

CODE	COURSE NAME	CATEGORY	L	T	P	CREDIT
20MCA246	MAIN PROJECT	PROJECT	-	-	27	12

Preamble: This project work aims to enable the students to apply the software engineering principles on a real software project, to make the students familiar with the stages of a deployment pipeline and to develop a software product using the latest software development methodology.

Prerequisite: Knowledge in software engineering principles and programming skills.

Course Outcomes: After the completion of the course the student will be able to

CO No.	Course Outcome (CO)	Bloom's Category Level
CO 1	Identify a real-life project which is useful to society / industry	Level 2: Understand
CO 2	Interact with people to identify the project requirements	Level 3: Apply
CO 3	Apply suitable development methodology for the development of the product / project	Level 3: Apply
CO 4	Analyse and design a software product / project	Level 4: Analyse
CO 5	Test the modules at various stages of project development	Level 5: Evaluate
CO 6	Build and integrate different software modules	Level 6: Create
CO 7	Document and deploy the product / project	Level 3: Apply

Mapping of course outcomes with program outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	2	3	3	3	1	2	3	3	3	3	3	3
CO 2	2	3	2	3	2	3	2	1	3	2	3	
CO 3	3	3	3	3	3	1	3	3	1		2	
CO 4	3	3	3	3	3		3	3	1	1	2	
CO 5	3	3	3	3	3		2	3			1	
CO 6	3	3	3	3	3		3	3		2	3	3
CO 7	1	1	3	3	3	2	3	3	2	1	2	

Mark distribution

Total Marks	CIE	ESE
100	70	30

Marks Division

Continuous evaluation by Supervisor, Guide(s) and Scrum Master	30 Marks (Internal)
Evaluation by the Project Assessment Board	40 Marks (Internal)
Evaluation by the External expert	30 Marks (External)
Total	**100 Marks**

Guidelines:

- Students shall identify Real-Life Projects which are relevant and useful to the society or industry.

- The project shall be an individual project and must be done in-house. The student has to spend time in the lab for the project work. Attendance as per MCA regulations is applicable for submitting the project for final evaluation.

- However, in exceptional cases students shall be given permission to work on the project outside the campus and at the industry premises if the organization offering the project belongs to anyone of the following categories.
  - CMM Level 5 Certified Company
  - Publicly listed company in India
  - National Research Institute
  - Central / State Government Department
  - Project funded by the Central / State Government Agency

- In such cases, the student is required to produce a letter from the organisation before starting the project and a committee constituted by the head of the department shall make the decision on permission. Industries and training institutes that offer project work for a fee shall not be permitted.

- Students shall submit project synopsis and get prior approval from the Project (Faculty) Supervisor before the project work begins.

- If there is a customer for the project then he/she will be the Product Owner (External Guide) and a faculty from the department will be the Internal Guide. If there is no such customer then the Internal Guide himself/herself shall act as the Product Owner.

- A faculty / technical staff shall act as the Scrum Master to continuously monitor the project development. Periodic meetings, of less than 15 minutes, at the convenience of the Scrum Master are to be highly encouraged. Ensure such meetings occur once in three days.

- The student shall maintain a Scrum Book (Rough Record) which has to be divided into 4 parts – (i) Product Backlog (ii) Database & UI Design (iii) Testing & Validation and (iv) Details of Versions. Make dated entries in the corresponding part at regular intervals. The corrections and comments from Product Owner and Scrum Master should be clearly indicated with the Date.

- Test Driven Development methodology may be practiced for the project development. BugZilla, BackLog or any such tool may be used for Bug Tracking.

- Git shall be used for Version Control and Git commit history may be verified as part of project evaluation .

- LaTeX or an equivalent tool shall be used for preparing Presentations and Project Report.

- Students shall be encouraged to publish their work in journals and due credit shall be given to such students.

- For the externally done projects, periodic confidential progress report and attendance statement shall be collected from the External Guide and be reviewed by the Project Supervisor.

- Set a sprint as two weeks, ensure biweekly reviews. A review shall not exceed 30 minutes. A demo to the Product Owner (Project Guide) is mandatory in every review.

- Interim evaluations of the project's progress shall be conducted by a Project Assessment Board as part of internal assessment. Two such evaluations are desirable. Scrum reviews shall not be sacrificed for such presentations.

- The Project Assessment Board shall be constituted by the Head of the Department with the following five members.
  Chairman:
    1. Head of the Department
  Members:
    2. Project supervisor/s of the student

3. One faculty member from the Department
4. One faculty member from a sister Department
5. An external expert, either from an academic/research institute or Industry. (For the externally done projects, the external guide shall be invited as external expert.)

- At the end of the semester, two evaluations shall be there on the entire project development activities. First an internal evaluation by the Project Assessment Board and second an external evaluation by an External Examiner.

- An External Examiner either from an academic institute or industry shall be appointed by the University for the External Evaluation.

Week	**Schedule**
(May be scheduled inline with the KTU academic calendar)	
1	Selection of Topic, Submission of project synopsis and getting approval Meeting of Development Team including Scrum Master with Product Owner (Project Guide)
2	Commencement of the Project.
4	First Sprint release and Scrum Review by the Product Owner (Project Guide)
6	Second Sprint release and Scrum Review by the Project Guide First interim evaluation by the Project Assessment Board
8	Third Sprint release and Scrum Review by the Project Guide
10	Fourth Sprint release and Scrum Review by the Project Guide
11	Second interim evaluation by the Project Assessment Board
12	Fifth Sprint release and Scrum Review by the Project Guide
13	Submission of project report, with Scrum Book Final project presentation Evaluation by the Project Assessment Board
14	Final evaluation by the External Examiner.