ST.JOSEPH'S
COLLEGE OF ENGINEERING
AND TECHNOLOGY,
- PALAI -

AUTONOMOUS

SJCET MCA Curriculum 2024

Choondacherry PO, Palai
Kottayam-686579, Kerala, India
Phone: 04822-239700, 239301, 239302
Website: www.sjcetpalai.ac.in, Email: info@sjcetpalai.ac.in

Signed By:Devassia V P
Approved by SJCET Academic Council held on 29.11.2024

ST. JOSEPH’S COLLEGE OF ENGINEERING AND TECHNOLOGY, PALAI (Autonomous)
DEPARTMENT OF COMPUTER APPLICATIONS
MASTER OF COMPUTER APPLICATIONS

CURRICULUM - SEMESTERS I TO IV
YEAR: 2024

SEMESTER I

Course No	Course	Hours/week			IA Marks	ESE Marks	Total	Credits	Exam Slot
		L	T	P					
24SJMCA101	Mathematical Foundations for Computing	3	1	-	40	60	100	4	A
24SJMCA103	Digital Fundamentals & Computer Architecture	3	1	-	40	60	100	4	B
24SJMCA105	Advanced Data Structures	3	1	-	40	60	100	4	C
24SJ MCA107	Advanced Software Engineering	3	1	-	40	60	100	4	D
24SJ MCA131	Programming Lab	0	1	3	50	50	100	2	R
24SJ MCA133	Web Programming Lab	0	1	3	50	50	100	2	S
24SJMCA135	Data Structures Lab	0	1	3	50	50	100	2	T
24SJ MCANC1	Entrepreneurship & Innovations in Technology	-	-	1	-	-	-	0	
		12	**7**	**10**	**310**	**390**	**700**	**22**	

SEMESTER II

Course No	Course	Hours/week			IA Marks	ESE Marks	Total	Credits	Exam Slot
		L	T	P					
24SJMCA102	Advanced Database Management Systems	3	1	-	40	60	100	4	A
24SJMCA104	Advanced Computer Networks	3	1	-	40	60	100	4	B
24SJMCA1--	Elective 1	3	1	-	40	60	100	4	C
24SJMCA1--	Elective 2	3	1	-	40	60	100	4	D
24SJMCA132	Object Oriented Programming Lab	0	1	3	50	50	100	2	R
24SJMCA134	Advanced DBMS Lab	0	1	3	50	50	100	2	S
24SJMCA136	Networking & System Administration Lab	0	1	3	50	50	100	2	T
24SJMCANC2	Industrial Readiness Training	-	-	1	-	-	-	0	
		12	**7**	**10**	**310**	**390**	**700**	**22**	

SEMESTER III

Course No	Course	Hours/week			IA Marks	ESE Marks	Total	Credits	Exam Slot
		L	T	P					
24SJMCA201	Data Science & Machine Learning	3	1	-	40	60	100	4	A
24SJMCA203	Design & Analysis of Algorithms	3	1	-	40	60	100	4	B
24SJMCA2--	Elective 3	3	1	-	40	60	100	4	C
24SJMCA2--	Elective 4	3	1	-	40	60	100	4	D
24SJMCA241	Data Science Lab	0	1	3	50	50	100	2	R
24SJMCA243	Mobile Application Development Lab	0	1	3	50	50	100	2	S
24SJMCA245	Mini Project	0		4	100	-	100	2	T
24SJMCANC3	Domain Expertise Workshops	-	-	1	-	-	-	0	
		12	**6**	**11**	**360**	**340**	**700**	**22**	

SEMESTER IV

Course No	Course	Hours/week			IA Marks	ESE Marks	Total	Credits	Exam Slot
		L	T	P					
24SJMCA242	Comprehensive Viva	-	-	-	-	100	100	6	
24SJMCA244	Seminar	-	-	2	50	-	50	2	
24SJMCA246	Main Project/Internship	-	-	27	70	30	100	12	
				29	**120**	**130**	**250**	**20**	
							2350	**86**	

Details Electives

Elective I	
24SJMCA162	Applied Statistics
24SJMCA164	Organizational Behaviour
24SJMCA166	Functional Programming
24SJMCA168	Virtualization and Containers
24SJMCA172	Advanced Operating Systems
Elective II	
24SJMCA182	Business Management
24SJMCA184	Embedded Systems
24SJMCA186	Computer Graphics
24SJMCA188	Artificial Intelligence
24SJMCA192	IPR and Cyber Laws
Elective III	
24SJMCA261	Operations Research
24SJMCA263	Cyber Security & Cryptography
24SJMCA265	Cloud Computing
24SJMCA267	Cyber Forensics
24SJMCA269	Compiler Design
Elective IV	
24SJMCA281	Internet of Things
24SJMCA283	Deep Learning
24SJMCA285	Digital Image Processing
24SJMCA287	Bioinformatics
24SJMCA289	Social Network Analysis

ST.JOSEPH'S
COLLEGE OF ENGINEERING
AND TECHNOLOGY,
- PALAI -

AUTONOMOUS

Choondacherry P.O., Pala, Kottayam - 686579
Kerala, India

SYLLABUS

MASTER OF COMPUTER APPLICATIONS

MCA

2024 SCHEME

www.sjcetpalai.ac.in

Programme Outcomes (POs)

PO1(Foundation Knowledge): Apply knowledge of mathematics, programming logic, and coding fundamentals for solution architecture and problem-solving.

PO2 (Problem Analysis): Identify, review, formulate, and analyze problems primarily focusing on customer requirements using critical thinking frameworks.

PO3 (Development of Solutions): Design, develop, and investigate problems with an innovative approach for solutions incorporating ESG/SDG goals.

PO4 (Modern Tool Usage): Select, adapt, and apply modern computational tools, such as the development of algorithms, with an understanding of the limitations, including human biases.

PO5 (Individual and Teamwork): Function and communicate effectively as an individual or team leader in diverse and multidisciplinary groups, using methodologies such as agile.

PO6 (Project Management and Finance): Apply the principles of project management, including scheduling and work breakdown structure, and be knowledgeable about finance principles for profitable project management.

PO7 (Ethics): Commit to professional ethics in managing software projects, especially in financial aspects. Learn to use new technologies for cybersecurity and insulate customers from malware.

PO8 (Life-long Learning): Continuously enhance management skills and the ability to learn, keeping up with contemporary technologies and ways of working.

INDEX

CURRICULUM

SEMESTER I

Course No	Course	Hours/week			IA Marks	ESE Marks	Total	Credits	Exam Slot
		L	T	P					
24SJMCA101	Mathematical Foundations for Computing	3	1	-	40	60	100	4	A
24SJMCA103	Digital Fundamentals & Computer Architecture	3	1	-	40	60	100	4	B
24SJMCA105	Advanced Data Structures	3	1	-	40	60	100	4	C
24SJ MCA107	Advanced Software Engineering	3	1	-	40	60	100	4	D
24SJ MCA131	Programming Lab	0	1	3	50	50	100	2	R
24SJ MCA133	Web Programming Lab	0	1	3	50	50	100	2	S
24SJMCA135	Data Structures Lab	0	1	3	50	50	100	2	T
24SJ MCANC1	Entrepreneurship & Innovations in Technology	-	-	1	-	-	-	0	
		12	7	10	310	390	700	22	

SEMESTER II

Course No	Course	Hours/week			IA Marks	ESE Marks	Total	Credits	Exam Slot
		L	T	P					
24SJMCA102	Advanced Database Management Systems	3	1	-	40	60	100	4	A
24SJMCA104	Advanced Computer Networks	3	1	-	40	60	100	4	B
24SJMCA1--	Elective 1	3	1	-	40	60	100	4	C
24SJMCA1--	Elective 2	3	1	-	40	60	100	4	D
24SJMCA132	Object Oriented Programming Lab	0	1	3	50	50	100	2	R
24SJMCA134	Advanced DBMS Lab	0	1	3	50	50	100	2	S
24SJMCA136	Networking & System Administration Lab	0	1	3	50	50	100	2	T
24SJMCANC2	Industrial Readiness Training	-	-	1	-	-	-	0	
		12	7	10	310	390	700	22	

ST.JOSEPH'S
COLLEGE OF ENGINEERING
AND TECHNOLOGY,
- PALAI -
AUTONOMOUS

MCA Regular

SEMESTER 1

24SJMCA101	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:		**K Level**
CO1	Understand mathematical reasoning in order to read, comprehend and construct mathematical arguments	K2
CO2	Count or enumerate objects and solve counting problems and analyze algorithms	K2
CO3	Solve problems in almost every conceivable discipline using graph models	K2
CO4	Solve the linear system of equations and calculate the eigen values and eigen vectors of matrices.	K3
CO5	Apply the principles of correlation and regression in practical problems.	K3

Mapping of course outcomes with program outcomes

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**
CO1	3	3	3		1			1
CO2	3	3	3					1
CO3	3	3	3		1			1
CO4	3	3	3					1
CO5	3	3	3					1

Mark distribution

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 $f, g: R \to R$ 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 $a_{n+2} - 4a_{n+1} + 3a_n = -200, n \geq 0$ given that $a_0 = 3000, a_1 = 3300$ $(K5)$

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 formprinciple 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**
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**
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**
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 grap	1
Euler circuit and trail, Planar and Non-Planar Graph	2
Module 4	**11**
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**
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

24SJMCA103	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:		**K Level**
CO1	Apply the basics of digital electronics to design and realize simple combinational logic circuits	K3
CO2	Apply the digital electronics principles to design sequential logic circuits.	K3
CO3	Understand the different design features of computer architecture, Five key components of a computer, processor and memory making technologies, addressing modes & instruction formats	K2
CO4	Understand Processor logic design conventions and data path, pipelining and hazards, I/O organization, Interrupts and direct memory access	K2
CO5	Understand and different types of memories - RAM, ROM, Cache memory, virtual memory etc. Apply the different memory design techniques.	K3

Mapping of course outcomes with program outcomes

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**
CO1	3	2	1					
CO2	3	3	1					
CO3	3	2		1				
CO4	3	3	1	1				
CO5	3	1	2	1				

Mark distribution

Continuous Internal Evaluation Pattern:

Attendance	: 8 marks
Continuous Assessment Test (2numbers)	: 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 Outcome (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.
2. List different types of pipeline hazards with examples.
3. 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.

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

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

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	**11hrs**
1.1	Representation of signed numbers – 1's complement and 2's complement, Logic gates - AND, OR, NOT, NAND, NOR, XOR	2
1.2	Boolean algebra - Basic laws and theorems, Boolean functions - truth table.	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
1.4	Floating point numbers	1
1.5	Combinational Circuits - Half adder - Full Adder	2
1.6	Decoder – Encoder - Multiplexers – Demultiplexers	2
	Module 2	**10hrs**
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	**10hrs**
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	**9hrs**
4.1	The Processor - Introduction, Logic design conventions, Building a Datapath, A simple implementation schemes.	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
	Module 5	**8hrs**
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

24SJMCA105	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:		**K Level**
CO1	Remember the Basic Data Structures and understand the Set Data Structure and its implementation.	K2
CO2	Understand Advanced Tree Structures for the design of efficient algorithms.	K3
CO3	Understand Advanced Heap Structures suitable for solving computational problems involving optimization and analysing these data structures using amortized analysis.	K3
CO4	Understand Advanced Graph algorithms suitable for solving advanced computational problems.	K2
CO5	Understand the basic operation of blockchain along with the data structures used in it and the challenges in blockchain data.	K2

Mapping of course outcomes with program outcomes

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**
CO1	3	1	2	1				
CO2	1	3	3	2				
CO3	1	3	3	3				
CO4	1	3	3	3				
CO5	1	3	3	3			2	

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 (COl)

(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 Amortized 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.

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

Course Contents and Lecture Schedule

No	Topic	**No. of Lecture Hours**
1	**Review of basic data structures**	**10Hrs**
1.1	Array, Stack and Queue Linked list and its variants	1
1.2	Representation of sets, Set implementation using bit string.	1
1.3	Hashing – SImple hash functions	1
1.4	Collision and Collision Resolution techniques	1
1.5	Amortised Analysis	1
1.6	Aggregate Method (Multipop Stack and Incrementing Binary Counter)	1
1.7	Accounting Method (Multipop Stack and Incrementing Binary Counter)	1
1.8	Potential Method (Multipop Stack and Incrementing Binary Counter)	1
1.9	Disjoint sets- representations	1
1.10	Union, Find algorithms	1
2	**Advanced Tree Structures**	**10Hrs**
2.1	Balanced Binary Search trees	1
2.2	Red-Black trees	1
2.3	Properties of Red Black trees	1
2.4	Rotations	1
2.5	Insertion	1
2.6	Deletion	1
2.7	B-Trees	1
2.8	Insertion and Deletion	1
2.9	Splay Trees	1
2.10	Suffix Trees	1
3	**Advanced Heap Structures**	**8Hrs**
3.1	Mergeable Heaps	1
3.2	Operations on Mergeable Heaps	1

3.3	Binomial Heaps	2
3.4	Binomial Heaps operations and Analysis	1
3.5	Fibonacci Heaps	2
3.6	Fibonacci Heap operations and Analysis.	1
4	**Advanced Graph Structures**	**12Hrs**
4.1	Representation of graphs	1
4.2	Depth First and Breadth First Traversals	2
4.3	Topological Sorting	1
4.4	Strongly connected Components	1
4.5	Biconnected Components	1
4.6	Minimum Cost Spanning Tree	1
4.7	Prim's Algorithm	1
4.8	Kruskal’s Algorithm	2
4.9	Dijikstra’s single source shortest paths algorithm	2
5	**Blockchain Data Structure**	**8Hrs**
5.1	Blockchain Architecture	1
5.2	Blockchain Data Structures	2
5.3	Blockchain Data types	2
5.4	Contract Data	2
5.5	Problems to be solved in Blockchain data analysis	1

24SJMCA107	ADVANCED SOFTWARE ENGINEERING	CATEGORY	L	T	P	CREDIT
		GENERAL	3	1	0	4

Preamble:	Most programs on Computer Applications do not give due importance to teaching Software Engineering from 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 C, C++, Java, Python or PHP programming language.

Course Outcomes: After the completion of the course the student will be able to:		**K Level**
CO1	Understand Software Engineering principles, the SDLC, software models, project planning, and requirements engineering.	K2
CO2	Learn coding standards, documentation, and version control with Git. Ensure software quality across SDLC stages.	K2
CO3	Understand OOP concepts, apply Design Patterns, and write unit tests using frameworks like JUnit or unittest.	K3
CO4	Use Agile and Scrum frameworks. Gain knowledge of software testing, automation, and defect management.	K2
CO5	Manage software configurations, CI/CD pipelines, and deployment automation using tools like Ansible and Robot Framework.	K2

Mapping of course outcomes with program outcomes								
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**
CO1	3	3	2	2	1			1
CO2	3	2	2	3	2			2
CO3	3	2	3	3	2			2
CO4	2	3	3	2	3	1	2	2
CO5	2	1	3	3	2	2		2

Mark distribution

Continuous Internal Evaluation Pattern:

Attendance	: 8 marks
Continuous Assessment Test (2numbers)	: 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 students should answer any one. Each question can have a maximum of 2 sub-divisions and carry 6 marks.

Course Outcome 1 (CO1)

- Understand software development as an engineering process and its stages.
- Gain knowledge of the **Software Development Lifecycle (SDLC)** and software engineering models.
- Learn to elicit, model, and document software requirements.
- Prepare **Software Requirements Specifications (SRS)** using **Use Cases** and **User Stories**.

Course Outcome 2 (CO2)

- Learn industry-grade coding practices following **style guides** and **coding standards**.
- Work with **version control systems**, including Git operations like **branching, cloning, committing, and conflict resolution**.
- Understand software quality concepts at different SDLC stages with a focus on documentation and conformance.

Course Outcome 3 (CO3)

- Comprehend **Object-Oriented Programming (OOP)** principles and apply **Design Patterns** effectively.
- Understand and implement **Unit Testing** using frameworks like **JUnit (Java)**, **unittest (Python)**, or **phpdbg (PHP)**.
- Gain insights into the **xUnit architecture** for writing effective tests.
- Learn the fundamentals of **Continuous Integration (CI)** and **Continuous Delivery (CD)** practices.

Course Outcome 4 (CO4)

- Understand the **Agile methodology** and implement the **Scrum framework** for software development.
- Learn essential **software testing methods**: **black-box testing**, **white-box testing**, and **regression testing**.
- Automate testing processes using tools like the **Robot Framework** to ensure quality

software delivery.

Course Outcome 5 (CO5)

- Gain knowledge of **Software Configuration Management (SCM)**, including dependency and build management.
- Understand **CI/CD pipeline** practices for continuous integration and delivery.
- Use **Ansible** for deployment automation and **Robot Framework** for test automation to streamline software releases.

SYLLABUS

Module I (8 Hours)

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 II (10 Hours)

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 III (10 Hours)

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 IV (10 Hours)

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 nonfunctional requirements.

Module V (10 Hours)

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.

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 July2020 [Module 2]

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

5. The Scrum Guide https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-

GuideUS.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]

References

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]

Course Contents and Lecture Schedule

No	Topic	No. of Lectures
1	**Module 1- Software Engineering**	**8**
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
	Module 2- Industry Best Practices	**10**

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
2.5	Software quality, achieving	2
	Module 3- System Design Methodologies	**10**
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
	Module 4- Agile Development Methodology	**10**
4.1	Agile Development methodology, introduction	2
4.2	Scrum framework	5
4.3	Automated testing	3
	Module 5- Continuous Integration and Continuous Development (CI/CD)	**10**
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

24SJMCA131	**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:		**K Level**
CO1	Understands basics of Python Programming language including input/output functions, operators, basic and collection data types	K2
CO2	Implement decision making, looping constructs and functions	K3
CO3	Design modules and packages - built in and user defined packages	K3
CO4	Implement object-oriented programming and exception handling.	K3
CO5	Create files and form regular expressions for effective search operations on strings and files.	K3

Mapping of course outcomes with program outcomes

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**
CO1	3	2		2				
CO2	3	2		2				
CO3	2	2	1	2				
CO4	3	2	1	2				
CO5	3	2		2				

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

Cycle 1

Course Outcome 1 (CO1): *Understands basics of Python Programming language including input/output functions, operators, basic and collection data types*

List of Experiments

1. Program to read Day, Month, and Year. Then display the corresponding date in DD-MM- YYYY format. Also, Check whether the date is from a leap or not.
2. Program to read a line of text. Count and display the occurrences of each word in a line of text. Take the first word in the text and exchange the first and last letters of the same.
3. Program to read three numbers and find the biggest. Let ‘n’ is the biggest number, compute n+nn+nnn. Assume the same ‘n’ is the radius of a circle and find its Area and Perimeter. Find the volume of a sphere with a radius ‘n’.
4. Create a list of colors (color list1) from comma-separated color names entered by the user. Display the first and last colors. Read another color list and print out all colors from color- list1 not contained in color-list2. Create another List from color list 1 by assigning an integer value for each color. From this list of integers, create a list removing even numbers.
5. Create a Dictionary. Sort the dictionary in ascending and descending order. Create another dictionary and Merge these two dictionaries.

Cycle 2

Course Outcome 2 (CO2): *Implement decision making, looping constructs and functions*

List of Experiments

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.

4. Count the number of characters (character frequency) in a string.
   - Add 'ing' at the end of a given string. If it already ends with 'ing', then add 'ly'
   - Accept a list of words and return length of longest word.

5.Generate all factors of a number.,Write lambda functions to find area of square, rectangle and triangle.

Cycle 3

Course Outcome 3 (CO3): *Design modules and packages - built in and user defined packages*

List of Experiments

1. Program to demonstrate 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.
3. Write programs that finds area and perimeter of figures by different importing statements. (Include selective import of modules and import * statements)

Cycle 4

Course Outcome 4 (CO4): *Implement object-oriented programming and exception handling*

List of Experiments

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 class for 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

Cycle 5

Course Outcome 5 (CO5): *Create files and form regular expressions for effective search operations on strings and files.*

List of Experiments

1. 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.
2. 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.
3. Write a Python program to read specific columns of a given CSV file and print the content of the columns.
4. 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

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

24SJMCA133	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:		**K Level**
CO1	Explore markup languages features and create interactive web pages using them.	K2
CO2	Learn and design client-side validation using scripting languages.	K2
CO3	Design front end web page and connect to the back-end databases.	K3
CO4	Do Client-side & Server-side scripting	K3
CO5	Develop Web Applications	K3

Mapping of course outcomes with program outcomes

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**
CO1	3	1	2					
CO2	3	2	3					
CO3	3		2					
CO4	3	1	2					
CO5	3	2	2	1				

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.

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

24SJMCA135	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 and knowledge of any programming language, preferably ‘C’.

Course Outcomes: After the completion of the course the student will be able to:		**K Level**
CO 1	Use Debuggers, Profilers and advanced Compiler options.	K3
CO 2	Implement the Set and Disjoint Set Data Structures.	K3
CO 3	Understand the practical aspects of Advanced Tree Structures.	K2
CO 4	Realize Modern Heap Structures for effectively solving advanced Computational problems.	K3
CO 5	Implement Advanced Graph algorithms suitable for solving advanced computational problems.	K3

Mapping of course outcomes with program outcomes

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**
CO1	3	2	2	3	1	1		2
CO2	2	3	3	2	1			1
CO3	2	3	3	3	1			2
CO4	2	3	3	3	1	1		2
CO5	2	3	3	3	2		2	2

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 (COI)

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 (COS)

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 24SJMCA105 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

Sl No	Topic	No. of hours
1	Advanced use of gcc : Important Options -o, -c, -D, -1, -I, -g, -O, -save-temps, -pg	1
2	Familiarization 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 operations9 (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

ST.JOSEPH'S
COLLEGE OF ENGINEERING
AND TECHNOLOGY,
- PALAI -
AUTONOMOUS

MCA Regular

SEMESTER 2

24SJMCA102	ADVANCED DATABASE MANAGEMENT SYSTEMS	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:		**K Level**
CO1	Understand the fundamentals of relational database systems including data models, database architectures and ER features.	K2
CO2	Analyze and apply the different normalization techniques.	K3
CO3	Assess the basic issues of transaction processing and concurrency control.	K3
CO4	Understand the roles that databases play in organizations and familiarize with basic database storage, file organization, database accessing techniques.	K2
CO5	Understand the basics of query processing, object-oriented and distributed databases, and analyze non-relational database systems, structures, and XML.	K2

Mapping of course outcomes with program outcomes

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**
CO1				1			1	
CO2	3	3	3	2			2	2

CO3	1	2	2	2		2		
CO4					1		1	
CO5	2			1				

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)
3. Explain the concept of XML. (K2)
4. Describe the various NoSQL databases. (K2)

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", 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

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
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
5	***Module V: System Architecture, Object Oriented Databases, XML hrs and NoSQL***	***13***
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	Non-relational Databases – MongoDB Sharding and Replication	1 hr
5.11	HBase	1 hr
5.12	Cassandra	1 hr

24SJMCA104	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 systems

Course Outcomes: After the completion of the course the student will be able to:		**K Level**
CO1	Comprehend the terminology and concepts of basic communication model, analyze the protocol layers and design application layer protocols.	K2
CO2	Understand and analyze the various transport layer protocols.	K2
CO3	Compare and contrast various routing algorithms in the network layer.	K2
CO4	Understand and analyze the concepts of link layer and physical layer.	K2
CO5	Understand how modern cellular and wireless networks work.	K2

Mapping of course outcomes with program outcomes

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**
CO1	3	2	2	1			1	1
CO2	3	3	2					
CO3	3	3	2					
CO4	3	2	2	1				
CO5	3	1	2	1				

Mark distribution

Continuous Internal Evaluation Pattern:

Attendance	: 8 marks
Continuous Assessment Test (2numbers)	: 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 Outcome (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.

Syllabus

Module I (10 Hours)

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, Web, HTTP, FTP, SMTP, POP3, and DNS, Peer-to-peer file sharing networks.

Module II (10 Hours)

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.

Module III (10 Hours)

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, software defined networks, Open flow.

Module IV (9 Hours)

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.

Module V (8 Hours)

IEEE 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 , Configuration management.

Text Books

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).

References

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
	Module 1	**10**
1.1	Overview of Computer Networks and the Internet History, Protocols, Review of last mile technologies used for internet access.	2
1.2	Packet switching, Basic ideas about delay, queuing, throughput. Concept of Quality of Service.	2
1.3	Protocol layering. OSI model and TCP model	2
1.4	Application layer protocols - Client-server architecture Network application architecture	2
1.5	Web, HTTP, FTP, SMTP, POP3 and DNS, Peer to-peer file sharing networks	2
	Module 2	**10**
2.1	Transport Layer Protocols: Introduction to transport layer	2
2.2	Multiplexing and demultiplexing, Principles of Reliable data transfer.	2
2.3	Stop-and-wait and Go-back- N design and evaluation	2
2.4	Connection oriented transport TCP, Connection less transport UDP	2
2.5	Principles of congestion control -efficiency and fairness	2
	Module 3	**10**
3.1	Network Layer Protocols: Virtual circuits and datagrams	2
3.2	Principles of routing, internet protocol Ipv4 , NAT	2
3.3	Routing algorithms: Link-state and distance vector routing	2
3.4	Routing on the internet RIP, OSPF and BGP, Multicast routing	2
3.5	Introduction to IPV6 and software defined networks	2
	Module 4	**9**
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)	2
4.3	IEEE 802.3, Ethernet, Switching and bridging, Media	2
4.4	Signal strength and interference. Data encoding.	1
4.5	Ethernet switches , Routers MAC, ARP, FIB	2
	Module 5	**8**
5.1	IEEE 802.11 Wi-Fi, Bluetooth, and cellular networks	2
5.2	Threats and attacks, Firewalls, NAT,VPNs	2
5.3	Introduction to network management, SNMP	2
5.4	Overview of tools and troubleshooting, Traffic analysis tools, configuration management.	2

24SJMCA172	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:		**K Level**
CO1	Identify synchronization problems in operating systems and issues in distributed systems.	K2
CO2	Explain classification of mutual exclusion algorithms and security violations.	K2
CO3	Explain the design of distributed shared memory and issues in load distribution.	K3
CO4	Explain design issues and synchronization in multiprocessor systems.	K3
CO5	Explain synchronization and concurrency control in database systems.	K3

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	1	2					
CO2	3	2	1				1	
CO3	3	1	2					
CO4	3	2	2					
CO5	3	1	1					

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 contains10 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.

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
	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
	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
	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 OperatingSystems*** – 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***”, PrenticeHall of India, 2007.
2. George Coulouris, Jean Dollimore,Tim Kindberg, Gordon Blair, ***“DistributedSystems,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**
	Overview: Functions of Operating System –Design Approaches –Types of Advanced Operating Systems.	2
	Synchronization Mechanisms: Concept of Processes and Threads – The Critical Section Problem – Other Synchronization Problems:– Monitor –Serializer – Path Expressions.	4
	Distributed Operating Systems:- Issues in Distributed Operating System – Communication Networks And Primitives –Lamport’s Logical clocks – Causal Ordering of Messages	4
2		
	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

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		
	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

24SJMCA182	BUSINESS MANAGEMENT	CATEGORY	L	T	P	CREDIT
		ELECTIVE	3	1	-	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 bookkeeping 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:		**K Level**
CO1	Understand management as a process.	K2
CO2	Critically analyse and evaluate management theories and practices	K2
CO3	Perform planning and organising for an organisation	K3
CO4	Do staffing and related human resource development function	K3
CO5	Take proper decisions to get a competitive advantage and understand basic concepts in book keeping and accounting	K2

Mapping of course outcomes with program outcomes

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**
CO1	3				2			2
CO2	3	3			2			1
CO3	3		3		2			
CO4	3				3			
CO5	2				3	2		2

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

Course Outcome C01: 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 and the procedure of preparation of balance sheet with case example.

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 controllingBreak 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 Lecture Hours
1	Introduction to Management: Basic Managerial concepts	2 Hours

1.1	Levels of management, Managerial Skills.	2 Hours
1.2	Management roles	1 Hours
1.3	Management functions	2 Hours
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 Hours
2	Planning: Nature and importance of planning, types of plans - Steps in planning, Levels of planning - The Planning Process	2 **Hours**
2.1	MBO definition and process, SWOT Analysis, importance. .	2 Hours
2.2	Organising : Nature of organizing,-span of control in management, factors affecting span of control- authority and responsibility.	3 Hours
2.3	Organisation structure - Formal and informal, Types of organization structure line, line and staff, functional, divisional, project, matrix, virtual form of organisations	**2 Hours**
3	Staffing and related HRD Functions: meaning, nature, staffing process	2 Hours
3.1	Job analysis and manpower planning, job description and job specification	2 Hours
3.2	Recruitment & selection, selection process, Tests and interviews. Training and development - concept and methods	3 Hours
3.3	Performance appraisal - concept and methods.	2 Hours
4	Managerial Decision Making and controlling : Decision making – types of decisions, decision making process, Decision Making Tools.	**2 Hours**
4.1	Importance of controlling, Techniques of controlling- Break Even Analysis, Budgetary Control	2 Hours
4.2	Benchmarking –importance and limitations of benchmarking	2 Hours
4.3	Six Sigma importance, limitations and process of six sigma	2 Hours
4.4	Total Quality Management-.	2 Hours
4.5	Introduction to marketing management-Marketing mix- product life cycle	2 Hours

5	Book- Keeping and Accountancy -Elements of Double Entry -BookKeeping	2 Hours
5.1	Rules for journalizing -Ledger accounts –Cash book-	3 Hours
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 Hours
5.3	Introduction to Accounting packages.	2 Hours

24SJMCA132	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:		**K Level**
CO1	Understand object-oriented concepts and design classes and objects to solve problems	K3
CO2	Implement arrays and strings.	K3
CO3	Implement object-oriented concepts like inheritance, overloading, interfaces, packages, exception handling and multithreading.	K3
CO4	Develop applications to handle events using AWT.	K3
CO5	Develop applications using files and networking concepts	K3

Mapping of course outcomes with program outcomes

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**
CO1	3	2	1					
CO2	3	2	1					
CO3	3	1	1					
CO4	3	1	2	2			1	
CO5	3	1	2				1	1

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)

6. Program to Sort strings

7. Search an element in an array.

8. Perform string manipulations

9.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):

10. Area of different shapes using overloaded functions

11.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.

12.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.

13.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.

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

15.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.

16.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.

17. Write a user defined exception class to authenticate the user name and password.

18. Program to remove all the elements from a linked list.

19. Program to remove an object from the Stack when the position is passed as parameter.

20. Write a Java program to compare two hash set.

Course Outcome 4(CO4):

21. Program to find maximum of three numbers using AWT.

22. Implement a simple calculator using AWT components.

23. Develop a program to handle all mouse events and window events

24. Develop a program to handle Key events.

Course Outcome 5(CO5):

25. Program to list the sub directories and files in a given directory and also search for a file name.

26. Write a program to write to a file, then read from the file and display the contents on the console.

27. Write a program to copy one file to another.

28. Write a program to implement Client-server communication using Socket TCP/IP

29. Write a program to implement Client-server communication using Datagram Socket - 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, Socket Programming

Reference Books

1. Herbert Schildt, ***Java The Complete Reference***- Seventh Edn -McGraw-Hill Edition
2. C. Thomas Wu, ***An introduction to Object-oriented programming with Java***- Foruth Edn- Tata McGraw-Hill Publishing company Ltd.
3. Cay S. Hostmann and Gray Cornel, ***Core Java: Volume I Fundamentals***- Eight Edn-Sun Microsystems Press.
4. K. Arnold, J. Gozling-***The JAVA programming language***- Third Edn- Pearson Education.
5. Paul Deitel Harvey Deitel- ***Java, How to Program Programming with Java***-Tenth Edn- Pearson Publishing,
6. Timothy Budd, ***Understanding Object-oriented programming with Java***- Updated Edn- 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

24SJMCA134	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:		**K Level**
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.	K3
CO 2	Apply PL/SQL for processing databases.	K3
CO 3	Comparison between relational and non-relational (NoSQL) databases and the configuration of NoSQL Databases.	K2
CO 4	Apply CRUD operations and retrieve data in a NoSQL environment.	K3
CO 5	Understand the basic storage architecture of distributed file systems.	K2
CO 6	Design and deployment of NoSQL databases with real time requirements.	K3

Mapping of course outcomes with program outcomes

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**
CO1	3	2	2	3	1	1		2
CO2	2	3	3	2	1			1
CO3	2	3	3	3	1			2
CO4	2	3	3	3	1	1		2
CO5	2	3	3	3	2		2	2

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

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

Sl No	Topic	No. of hours
1	An overview of relational database design using MySQL/ MariaDB/ PostgreSQL etc.(Apply the following basic queries on an Employee/ Student database etc.)	6
	• DDL Commands • DML Commands • Imposing restrictions on database (DCL & TCL Commands)	3
	• 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
2	PL/SQL Programs	4
2.1	Trigger, Cursor, Stored Procedures and Functions.	4
3	Introduction to NoSQL Databases	2
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

5	Query Processing	8
5.1	- Performing CRUD operations - Retrieving Data from a NoSQL database - Usage of aggregate functions, regular expressions etc.	8
6	NoSQL Administration	2
6.1	- Security, Monitoring & Backup - Create Users and Roles	2
7	NoSQL shell commands	6
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
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

24SJMCA136	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:		**K Level**
CO1	Install and configure common operating systems.	K2
CO2	Perform system administration tasks.	K2
CO3	Install and manage servers for web applications.	K3
CO4	Write shell scripts required for system administration.	K3
CO5	Acquire skill sets required for a DevOps.	K3

Mapping of course outcomes with program outcomes

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**
CO1	3	1		1				
CO2	3	1	1					
CO3	3	1	1	1				
CO4	3	2	1					
CO5	3	1	1	2	1	1	1	

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. 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, interfacing 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

Department of

Computer Applications

Vision

To emerge as a center of excellence in the field of computer education with distinct identity and quality in all areas of its activities and develop a new generation of computer professionals with proper leadership, commitment, and moral values.

Mission

- ➢ Provide quality education in Computer Applications and bridge the gap between the academia and industry.
- ➢ Promote innovation, research, and leadership in areas relevant to the socio-economic progress of the country.
- ➢ Develop intellectual curiosity and a commitment to lifelong learning in students, with societal and environmental concerns.

ST.JOSEPH'S COLLEGE OF ENGINEERING AND TECHNOLOGY, - PALAI -

AUTONOMOUS

Vision

Developing into a world class, pace setting institute of Engineering and Technology with distinct identity and character, meeting the goals and aspirations of the society.

Mission

- To maintain a conducive infrastructure and learning environment for world class education.
- To nurture a team of dedicated, competent and research-oriented faculty.
- To develop students with moral and ethical values, for their successful careers, by offering variety of programs and services.