

**MANONMANIAM SUNDARANAR UNIVERSITY, TIRUNELVELI-12**

**AFFILIATED COLLEGES**

**MASTER OF SCIENCE in SOFTWARE ENGINEERING**

*Learning Outcome based Curriculum Framework (LOCF)*

**With effect from 2021-2022 onwards**

**VISION AND MISSION OF THE UNIVERSITY**

**Vision**

“To provide quality education to reach the un-reached”

**Mission**

- To conduct research, teaching and outreach programmes to improve conditions of human living
- To create an academic environment that honours women and men of all races, caste, creed, cultures and an atmosphere that values intellectual curiosity, pursuit of knowledge, academic freedom and integrity
- To offer a wide variety of off-campus educational and training programs, including the use of information technology, to individuals and groups.
- To develop partnership with industries and government so as to improve the quality of the workplace and to serve as catalyst for economic and cultural development
- To provide quality / inclusive education, especially for the rural and un-reached segments of economically downtrodden students including women, socially oppressed and differently abled

## **PROGRAM EDUCATIONAL OBJECTIVES**

1. To prepare the students to understand the core concepts in **SOFTWARE ENGINEERING**
2. Enable students to develop problem solving and programming skills in the recent technologies thereby developing strong employability
3. Empower students to prepare themselves to engage in active research
4. Enable students to pursue competitive exams at National and state level such as NET/SLET/GATE

## **PROGRAM OUTCOMES (POs)**

*On completion of the M.Sc. (Software Engineering) programme, the students will be able to:*

- PO1:** Identify and understand the need for basic mathematical and Computational Concepts and apply them to real world problems.
- PO2:** Design and develop softer applications using computers to analyze and solve real world problems.
- PO3:** Design, implement and evaluate a computer-based software, process, component, or programs to meet the stakeholder needs
- PO4:** Analyze, design and choose efficient algorithms and apply them in appropriate Context in the software development
- PO5:** Analyze large data sets in the context of real world problems and interpret results using data analytics.
- PO6:** Understand software testing methods and apply them to analyze data for decision Making.
- PO7** Realize the importance of lifelong learning and continuous professional development.

## **Programme Specific Outcomes (PSO)**

On Successful completion of the M.Sc. (Software Engineering) degree programme, students will be able to;

- PSO1:** Analyze, design and develop solutions to significant computational problems.
- PSO2:** Utilize tools and techniques necessary for developing Software Solutions.

**PSO3:** Participate in competitive exams such as SET, NET etc. effectively.

**PSO4:** Design, develop and evaluate Object oriented Software to meet the desired demands of industry and society.

**PSO5:** Demonstrate best practices and standards in the design of quality Software.

**PSO6:** Work with computing technologies and pursue career in the areas related to Software Engineering.

**PSO7:** Function effectively as an individual or in teams involving software design in multidisciplinary environments.

## REGULATIONS of the PROGRAMME

**Duration of the Programme:** Two years (4 Semesters)

**Eligibility:**

**Students with three year Bachelor's degree in Software Engineering / Computer Science / Computer Applications / Information Technology or any other degree accepted by the Syndicate of Monomania Sundaranar University as equivalent in the 10+2+3 pattern**

### SEMESTER WISE COURSE LIST

SEMESTER I					
Semester	Course No.	Course type	Course Name	Contact Hrs./ Week	Credits
I	1	Core-1	Design and Analysis of Algorithms	5	4
	2	Core-2	Advanced Java Programming	5	4
	3	Core-3	Mathematical Foundation for Computer Science	4	4
	4	Core-4	Advanced Software Engineering	4	4
	5	Core - 5	Distributed operating system	4	4
	6	Core - 6 Practical - 1	Algorithm Lab	4	2
	7	Core - 7 Practical - 2	Advanced Java Lab	4	2
Subtotal				30	24
SEMESTER II					
Semester	Course No.	Course type	Course Name	Contact Hrs./ Week	Credits
II	8	Core- 8	Advanced Web Technology	5	4
	9	Core- 9	Machine Learning	5	4

	10	Core- 10	Advanced DBMS	4	4
	11	Core- 11	Software Project Management	4	4
	12	Elective– 1 ( Select any one )	1.Free open source Software 2.Data Mining 3. Software Quality Assurance	4	3
	13	Core - 12 Practical - 3	Advanced Web Technology Lab	4	2
	14	Core - 13 Practical - 4	Machine Learning Lab using Python	4	2
	<b>Subtotal</b>			<b>30</b>	<b>23</b>

<b>SEMESTER III</b>					
<b>Semester ( 1 )</b>	<b>Course No. ( 2 )</b>	<b>Course Type ( 3 )</b>	<b>Course Name ( 4 )</b>	<b>Contact Hrs./Week ( 5 )</b>	<b>Credits ( 6 )</b>
III	15	Core-14	Software Testing	4	4
	16	Core-15	Soft Computing	4	4
	17	Core-16	Advanced Computer Networks	4	4
	18	Core-17	Research Methodology	4	4
	19	Elective - 2 ( Select any one )	1. Cloud Computing 2. Mobile Application Development 3. Object Oriented Software Engineering	4	3
	20	Core - 18 Practical - 5	Software Testing LAB	4	2
	21	Core –19	Mini Project	6	6
	<b>Subtotal</b>			<b>30</b>	<b>27</b>
IV	22	Core – 20	Major Project	30	16
	<b>Subtotal</b>			<b>30</b>	<b>16</b>
<b>Cumulative total</b>				<b>120</b>	<b>90</b>

**Scheme of Examination / Question Paper Pattern I - Theory Course:**

(Total Marks: 100 (Internal: 25 Marks, External: 75 Marks))

<b>Parameters</b>	
<b>Student shall secure pass in both internal and external and also obtain 50 marks together to get a pass</b>	
<b>CIA- Internal Marks</b>	<b>End semester Examination - External Marks</b>
i. Average of best two tests from three: 15 Marks	Total : 75 Marks
ii. Seminar: 05 Marks	
iii. Assignment: 05 Marks -----	
Total : 25 Marks	
Passing minimum 40% i.e. 10 marks	Passing minimum 50% i.e. 38 marks

**External (End Semester) examination question pattern:**

Time: 3 Hours	Max. Marks: 75
<b>Part – A</b>	
(10*1=10)	
<i>Answer all the questions</i>	
Ten Questions, two objective type questions from each unit.	
<b>Part – B</b>	
(5*5=25)	
<i>Answer all the questions</i>	
Five Questions, two short answer type questions from each unit with internal choice ( Either ...Or... type)	
<b>Part – C</b>	
(5*8=40)	
<i>Answer all the questions</i>	
Five Questions, two descriptive/Analytical type questions from each unit with internal choice ( Either ...Or... type)	

## **Practical Courses**

**Assessment Components (External: Internal (CIA) – 50: 50)**

### **Passing Criteria for Practical Examinations:**

There is **no passing minimum for the Continuous Internal Assessment (CIA)** component. But overall (CIA+ External), a student shall secure minimum of 50% or more to get a pass.

### **End Semester practical Examinations**

Practical examinations will be conducted at the end of each semester. The scheme of valuation is to be decided by the respective board of Question setters before the commencement of Practical exams.

### **Passing Minimum for Mini/Major Project:**

**There is no Passing Minimum for the CIA component of the Mini/Major Project.** But overall (CIA+ External), a student shall secure 50% or more to get a pass.

### **End Semester Mini/Major Project Examinations**

Mini and Major Project viva voce examinations will be conducted at the end of third and fourth semester respectively. Students are required to submit project reports before the viva voce examination.

**Mini and Major Project work Evaluation criteria:** Please Refer Semester III and Semester IV course details

## Semester I

Course No.	Course type	Course Name	Contact Hrs./ Week	Credits
1	Core-1	Design and Analysis of Algorithms	5	4
2	Core-2	Advanced Java Programming	5	4
3	Core-3	Mathematical Foundation for Computer Science	4	4
4	Core-4	Advanced Software Engineering	4	4
5	Core - 5	Distributed operating system	4	4
6	Core - 6 Practical - 1	Algorithm Lab	4	2
7	Core - 7 Practical - 2	Advanced Java Lab	4	2
<b>Subtotal</b>			<b>30</b>	<b>24</b>

## CORE - 1 DESIGN AND ANALYSIS OF ALGORITHMS [C L T P 4 4 1 0]

### Course Objectives:

1. Learn fundamental algorithmic design paradigms
2. Design algorithms using popularly used strategies like greedy, divide and conquer, dynamic programming etc.
3. Analyze the algorithmic procedure to determine the computational complexity of algorithms

### Course Prerequisites:

1. Exposure to introductory course on programming languages
2. Basic Concepts of Data Structures
3. Some Basic Mathematics

### Course Outcomes (COs):

After the completion of this course, student will be able to

**CO1:** Identify the Characteristics of an algorithm

**CO2:** Understand the problem-solving approaches using computers

**CO3:** Compute the time complexity of an algorithm

**CO4:** Analyze the performance of an algorithm in solving a problem

**CO5:** Compare the performance of various algorithms in solving a specific-problem

**CO6:** Develop algorithms for solving real-time problems

### Course Outline:

**TOTAL (60 HOURS)**

#### UNIT-1

**(12 HOURS)**

Introduction: Algorithm-Specification - Performance Analysis. Elementary Data Structure: Stacks and Queues- Trees-Dictionaries-Priority Queues- Graphs

#### UNIT – II

**(12 HOURS)**

Divide and Conquer: General Method-Binary Search- Finding the Maximum and Minimum-Quick sort - Strassen's Matrix Multiplication.

#### UNIT – III

**(12 HOURS)**

The Greedy Method: General Method-Knapsack Problem-Job Sequencing with Deadlines-

Minimum Cost Spanning Tree-Single Source Shortest Path. Dynamic Programming: General Method-Multistage Graph-All Pairs Shortest Path-Optimal Binary Search Tree-0/1 Knapsack-Travelling Salesperson Problem.

**UNIT – IV**

**(12 HOURS)**

Basic Traversal and Search Techniques: Techniques for Binary Trees –Techniques for Graphs-Connected Components and Spanning Trees-Biconnected Components and DFS. Backtracking: General Method-8-Queen Problem, Sum of Subsets Graph Coloring: Hamiltonian Cycle.

**UNIT – V**

**(12 HOURS)**

Branch and Bound: The Method-0/1 Knapsack Problem. NP-Hard And NP - Complete Problem - Basic Concepts - Cook’s Theorem -NP - HARD GRAPH Problems - Clique Decision Problem - Chromatic Number Decision Problem NP - HARD Scheduling Problems - Flow Shop Scheduling - Job shop scheduling.

**Mapping of COs to POs and PSOs**

Course Outcome	PO Addressed PO1 to PO7	Correlation Level L/M/H	PSO addressed PSO1 to PSO7	Correlation Level L/ M/ H	Cognitive Level K <sub>1</sub> to K <sub>6</sub>						
CO1	PO1	H	PSO1	H	K <sub>1</sub>						
CO2	PO1	M	PSO1	M	K <sub>2</sub>						
CO3	PO2	H	PSO2	M	K <sub>3</sub>						
CO4	PO4	H	PSO6	H	K <sub>4</sub>						
CO5	PO6	M	PSO3	PSO5	H	M	K <sub>5</sub>				
CO6	PO3	PO5	PO7	M	H	M	PSO5	PSO7	M	H	K <sub>6</sub>

(L – Low, M – Medium, H – High; K<sub>1</sub> – Remember, K<sub>2</sub> – Understand, K<sub>3</sub> – Apply, K<sub>4</sub> – Analyze, K<sub>5</sub> – Evaluate, K<sub>6</sub> – Create)

**Text and Reference books**

1. Ellis Horowitz, Sartaj Sahni and Sanguthevar Rajasekaran, —Fundamentals of Computer Algorithms, 2nd Edition, Universities Press (India) Private Ltd., 2018.
2. Aho, Hopcroft and Ullman, —The Design and Analysis of Computer Algorithm, Pearson Education, Delhi, 2001.
3. Basu S.K., —Design Methods and Analysis of Algorithms, PHI, 2006.
4. M.A.Weiss, —Data Structures and algorithm Analysis in C++, Pearson Education, Asia, 2013.
5. Sandeep Sen and Amit Kumar Design and Analysis of Algorithms: A contemporary perspective, Cambridge University Press, 2019.

## CORE- 2 ADVANCED JAVA PROGRAMMING [C L T P 4 4 1 0]

### Course Objectives:

1. To design stand-alone desktop-oriented GUI based Java applications using Swing and access the database using JDBC.
2. To understand the networking components to transfer data over networks.
3. To develop web application based on Java uses Servlet, JSP
4. To design applications using pre-built frameworks.

### Course Prerequisites:

1. Experience in any of the Object Oriented Programming language
2. Basic understanding of the Java language
3. Basic knowledge on SQL

### Course Outcomes (COs):

At the end of the Course, the student will be able to -

- CO1:** Understand swing components and its usage.
- CO2:** Implement Networking and Data base connectivity in Java for given application.
- CO3:** Implement webpage with dynamic content and server side web application using Servlet and JSP.
- CO4:** Develop Java application using spring framework.

**Course Outline** **Total (60 HOURS)**

**Unit I-Applets & Event Handling:** **(12 HOURS)**

**Applets** - Applet Basics, Methods of Building an Applet, Some General Methods of Applet , Displaying Text in Status Bar, Embedding Applet Information, The HTML Applet Tag , Reading Parameters into Applets , Colors in Applet, Getting Document base and Codebase, Interfaces in Applet, Multimedia in Applet **Event Handling**- Model, Event, Event Listeners, Registering Listener with Source, Example programs, Adapter Classes.

**Unit II – Swing and GUI Components:** **(12 HOURS)**

**Swing** - Origin of Swing, Creating Windows in Swing, JFrame,Jbutton, JLabel, JToggleButton, JCheckBox, JRadioButton, JList, JScrollBar, JScrollPane, JTextField, JPasswordField, JTextArea,

JComboBox, JMenuItem, JMenu, JMenuBar, JDialog, JOptionpane, JFileCheck, JProgressBar, Layout Manager.

**Unit III- Networking: (8 HOURS)**

**Networking** - InetAddress, Socket Programming, Datagram, URL.

**Unit IV- JDBC: (12 HOURS)**

**JDBC**- Introduction, Driver Manager, Connection Interface, Statement Interface, Prepared Statement Interface, Callable Statement Interface, Result Set Interface.

**Unit V - Servlet & JSP: (16 HOURS)**

**Servlet** – Introduction, HTML, Interface Servlet, HttpServlet Class, Servlet Programs, Servlet with I/O Files, Servlet with JDBC, Session Handling, Session Tracking. **JSP** – Introduction, JSP Working Model, Syntax of a JSP Page with Sample Programs.

**Mapping of COs to POs and PSOs**

Course Outcome	PO Addressed	Correlation Level		PSO Addressed		Correlation Level		Cognitive Level
				PSO1	PSO4	M	H	
CO1	PO2	H		PSO1	PSO4	M	H	K <sub>1</sub>
CO2	PO3	H		PSO2		H		K <sub>6</sub>
CO3	PO3	H	PSO4	H		K <sub>6</sub>		
CO4	PO2	H		PSO2		H		K <sub>3</sub>

(L – Low, M – Medium, H – High; K<sub>1</sub> – Remember, K<sub>2</sub> – Understand, K<sub>3</sub> – Apply, K<sub>4</sub> – Analyze, K<sub>5</sub> – Evaluate, K<sub>6</sub> – Create)

**Text and Reference Books**

1. Advanced Programming in Java2, K. Somasundaram, Jaico publishing Company Limited, New Delhi, 2008.
2. Herbert Scheldt, Java 2-The complete reference, 7<sup>th</sup> Edition McGraw Hill, 2018.
3. Naughton and Herbert Schildt, Java The complete reference, 7<sup>th</sup> Edition McGraw Hill, 2007.
4. Jim Keogh, The Complete Reference J2EE, Tata McGraw Hill Edition, New Delhi, 2002.
5. Marty Hall, and Larry Brown, Core Servlets and Java Server Pages, 2<sup>nd</sup> Edition, Pearson Education, 2004.

**E-Resources:**

1. Advanced Programming in Java2, [https://www.researchgate.net/publication/315894230\\_Advanced\\_Programming\\_in\\_Java2](https://www.researchgate.net/publication/315894230_Advanced_Programming_in_Java2)

2. JDBC, Java Database Connectivity, K.Somasundaram, Jaico Publishing House, Mumbai, India, First Edition, 2013.
3. JDBC Connectivity in Java JDK16, June 2021, DOI: 10.13140/RG.2.2.19415.60325  
[https://www.researchgate.net/publication/352172393 JDBC Connectivity in Java JDK16](https://www.researchgate.net/publication/352172393_JDBC_Connectivity_in_Java_JDK16)
4. Installing Eclipse 2019-12-R and Tomcat 9.0 and Develop a Servlet, June 2021, DOI: 10.13140/RG.2.2.12123.08487  
[https://www.researchgate.net/publication/352785295 Installing Eclipse 2019-12-R and Tomcat 90 and Develop a Servlet](https://www.researchgate.net/publication/352785295_Installing_Eclipse_2019-12-R_and_Tomcat_90_and_Develop_a_Servlet)
5. JSP, Java Server Pages, In book: Server Side Programming Chapter: Chapter 25, K.Somasundaram, 2012, DOI: 10.13140/2.1.1715.9365
6. Java Server Pages  
[https://www.researchgate.net/publication/268076772 Java Server Pages](https://www.researchgate.net/publication/268076772_Java_Server_Pages)

## CORE - 3 MATHEMATICAL FOUNDATION FOR COMPUTER SCIENCE

[CLTP 4310]

### Course Objectives:

- To introduce the concepts of sets and perform associated operations on them.
- To introduce the concepts of mathematical logic.
- To provide basic understanding on abstract algebraic structures
- To construct graphs, automata and discuss its applications
- To understand queuing systems and their classifications

### Course Prerequisites:

- Basic Knowledge in Mathematics
- Knowledge of the fundamental concepts in the undergraduate level

### Course Outcomes (COs):

After the completion of this course, student will be able to

**CO1:** Evaluate the validity of logical arguments and construct mathematical proofs

**CO2:** administer all the basic operations with sets

**CO3:** understand abstract algebraic structure like groups and their properties

**CO4:** Analyze whether given graphs are isomorphic and apply different algorithms to find the shortest path Apply Mathematical techniques into many areas of Computer science like Algorithms, Computer Networks, and Cryptography etc.

**CO5:** Apply Mathematical techniques into many areas of Computer science like Algorithms, Computer Networks, and Cryptography etc.

### Course Outline

**(Total 60 HOURS)**

#### UNIT-1

**(12 HOURS)**

Mathematical Logic: Introduction-Statements and Notation-Connectives-Normal Forms-The Theory of Inference for the Statement calculus –The Predicate Calculus-Inference theory of the Predicate Calculus

#### UNIT – II

**(12 HOURS)**

Set Theory: Introduction – Basic concepts of Set theory-Relations-Functions.

**UNIT – III****(12 HOURS)**

Matrices: Rank of Matrix, Solving System of Equations, Eigen values and Eigen Vectors-Inverse of matrix- Cayley Hamilton Theorem.

**UNIT – IV****(12 HOURS)**

Graph Theory: Introduction-Graphs-Application of Graphs-Finite and Infinite Graphs-Incidence and Degree-Isolated Vertex, Pendant Vertex and Null Graph Paths and Circuits: Isomorphism-Sub graphs - Walks, Paths and Circuits-Connected and Disconnected graphs-Components-Euler Graphs-Operation on Graphs -Hamiltonian paths and circuits- Travelling Salesman Problem

**UNIT – V****(12 HOURS)**

Trees and Fundamental Circuits: Properties of Trees-Pendant vertices in trees-Distance and centers in a Tree-Rooted and Binary Tree-Spanning Tree-Fundamental circuits—Spanning Tree in the Weighted Graph. Matrix Representation of Graphs: Incidence Matrix-Sub matrices-Circuit Matrix-Path matrix-Adjacency Matrix

**Mapping of COs to POs and PSOs**

Course Outcome	PO Addressed PO1 to PO7		Correlation Level L/M/H		PSO Addressed PSO1 to PSO7		Correlation Level L/ M/ H		Cognitive Level K <sub>1</sub> to K <sub>6</sub>
CO1	PO1		H		PSO6		H		K <sub>4</sub>
CO2	PO1		H		PSO6		M		K <sub>3</sub>
CO3	PO1		H		PSO1		H		K <sub>2</sub>
CO4	PO4		M		PSO4		H		K <sub>4</sub>
CO5	PO2	PO6	M	H	PSO2	PSO3	M	H	K <sub>3</sub>

(L – Low, M – Medium, H – High; K<sub>1</sub> – Remember, K<sub>2</sub> – Understand, K<sub>3</sub> – Apply, K<sub>4</sub> – Analyze, K<sub>5</sub> – Evaluate, K<sub>6</sub> – Create)

**Text and Reference books**

1. J.P. Trembley and R.Manokar, —Discrete Mathematical Structures with Applications to Computer Science, Tata McGraw Hill Publications, 2017.
2. Narasingh Deo, — Graph Theory with Applications To Engineering And Computer Science||, Prentice-Hall of India Private Limited, 2017.
3. Kenneth H. Rosen — Discrete Mathematics and Its Applications with Combinatorics and Graph Theory (SIE) | 7th Edition, Tata McGraw Hill Publications, 2017.

## **CORE - 4 ADVANCED SOFTWARE ENGINEERING [C L T P 4 3 1 0]**

### **Course Objectives:**

- To comprehend software engineering lifecycle models.
- To do project management and cost estimation.
- To advance in the knowledge of the System Analysis and Design concepts.
- To understand software testing approaches.
- To be familiar with software quality management.

### **Course Prerequisite**

Basic Knowledge of mathematics and data structure

### **Course Outcome**

After the completion of this course, student will be able to;

- CO1** Apprehend the leads of various Software Development Lifecycle Models
- CO2** Gain insight on project management approaches as well as cost and schedule estimation strategies.
- CO3** Conduct a detailed analysis of the requirements
- CO4** Use UML diagrams for ease and effective way to do analysis and design
- CO5** Be familiar with software testing methodologies.
- CO6** Smooth maintenance of software reliability and quality management.

### **Course Outline:**

**(Total 60 hours)**

#### **UNIT-1 Introduction**

**(12 hours)**

Emergence of Software Engineering - Notable Changes in Software Development Practices - Computer Systems Engineering - System Development Life Cycle (SDLC) - A Few Basic Concepts - Waterfall Model and its Extensions - Rapid Application Development (RAD) - Agile Development Models - Spiral Model

#### **UNIT – II Software Metrics**

**(12 hours)**

Software Project Management Complexities - Responsibilities of a Software Project Manager - Project Planning - Metrics for Project Size Estimation - Project Estimation Techniques - Empirical Estimation Techniques - COCOMO: A Heuristic Estimation Technique - Halstead's Software Science: An Analytical Technique - Staffing Level Estimation - Scheduling - Organisation and Team Structures - Staffing - Risk Management - Software Configuration Management

**UNIT – III Software Design****(12 hours)**

Overview of the Design Process - How to Characterize a Good Software Design? - Cohesion and Coupling - Layered Arrangement of Modules - Approaches to Software Design - Function-oriented Design - Overview of SA/SD Methodology - Structured Analysis - Developing the DFD Model of a System - Structured Design - Detailed Design - Design Review - Object-oriented Design - Basic Object-Orientation Concepts - Unified Modelling Language (UML) - UML Diagrams - Use Case Model - Class Diagrams – Interaction Diagrams - Activity Diagram - State Chart Diagram – Postscript - User Interface Design - Characteristics of a Good User Interface - Basic Concepts - Types of User Interfaces - Fundamentals of Component-based GUI Development - A User Interface Design Methodology.

**UNIT – IV Coding and Testing****(12 hours)**

Coding - Code Review - Software Documentation – Testing - Black-box Testing - White-Box Testing – Debugging - Program Analysis Tools - Integration Testing - Testing Object- Oriented Programs - System Testing - Some General Issues Associated with Testing

**UNIT – V Software Reliability and Quality Management****(12 hours)**

Software Reliability- Software Quality - Software Quality Management System - ISO 9000 - SEI Capability Maturity Model - Characteristics of Software Maintenance - Software Reverse Engineering - Software Maintenance Process Models - Estimation of Maintenance Cost - Emerging Trends - Client-Server Software - Client-server Architectures - CORBA - COM/DCOM - Service-Oriented Architecture (SOA) - Software as a Service (SaaS).

**Mapping of COs to POs and PSOs**

Course Outcome	PO Addressed PO1 to PO8		Correlation Level L/M/H		PSO Addressed PSO1 to PSO7		Correlation Level L/ M/ H		Cognitive Level K <sub>1</sub> to K <sub>6</sub>
CO1	PO1		M		PSO1		L		K <sub>1</sub>
CO2	PO1	PO2	L	M	PSO1	PSO3	M	H	K <sub>2</sub>
CO3	PO5	PO1	H	M	PSO1	PSO4	H	H	K <sub>3</sub>
CO4	PO2	PO4	H	M	PSO1		M		K <sub>4</sub>
CO5	PO4		H		PSO5		H		K <sub>5</sub>
CO6	PO6	PO7	H	H	PSO6		H		K <sub>6</sub>

(L – Low, M – Medium, H – High; K<sub>1</sub> – Remember, K<sub>2</sub> – Understand, K<sub>3</sub> – Apply, K<sub>4</sub> – Analyze, K<sub>5</sub> – Evaluate, K<sub>6</sub> – Create)

### ***Text books and References***

1. Rajib Mall, Fundamentals of Software Engineering, 5th edition, PHI Learning Pvt. Ltd., 2018.
2. Ian Sommerville, Software Engineering, 10th edition, Pearson, 2017.
3. David Thomas and Andrew Hunt, The Pragmatic Programmer: your journey to mastery, 20th Anniversary Edition, Pearson, 2019.
4. Jim Cooling, The Complete Edition – Software Engineering for Real-Time Systems: A software engineering perspective toward designing real-time systems, Packt, 2019.
5. Hitesh Mohapatra and Amiya Kumar Rath, Fundamentals of Software Engineering: Designed to provide an insight into the software engineering concepts, BPB Publications, 2020.
6. Achilleas Anagnostopoulos, Hands-On Software Engineering with Golang: Move beyond basic programming to design and build reliable software with clean code, Packt, 2020.

## **CORE - 5 DISTRIBUTED OPERATING SYSTEM**

**[C L T P 4 3 1 0]**

### **Course Objectives:**

To make the students to realize the importance of the operating system in the computing domain.

Emphasis would be to provide the knowledge of communication, synchronization, resource management and security aspect in distributed operating system

Explicitly define and intuitively describe why operating systems virtualize hardware and how the operating system makes it possible for many applications to share resources and to make programming easier for user space applications

Configure a Linux-based operating system and work from the shell

Understand the procedures to manage files and directories in the Linux operating system

Develop and debug systems software

### **Course Prerequisites:**

Knowledge of computer systems organization

### **Course Outcomes (COs):**

After the completion of this course, student will be able to

**CO1:** Gain knowledge about the history of the Linux operating system, its unique licensing model and the major distributions that are available to use

**CO2:** Start and stop services from running in the Linux operating systems.

**CO3:** Implement process scheduling algorithms

**CO4:** Learn to manage files and directories in the Linux operating system

**CO5:** To use the Linux environment for problem solving

### **Course Outline:**

**(Total 60 HOURS)**

#### **UNIT-1**

**(12 HOURS)**

Fundamentals: What is Distributed Operating System – Evolution of Distributed Computing System – Distributed Computing System Models – Why are Distributed Computing Systems gaining popularity – What is a Distributed Computing System – Issues in Designing Distributed Computing System – Introduction to Distributed Computing Environment. Introduction to Computer Networks – Network types – LAN –WAN – Communication protocols – Internetworking – ATM Technology

**UNIT – II****(12 HOURS)**

Message Passing: Introduction Desirable features – Issues in PC Message Passing – Synchronization – Buffering – Multi datagram Messages – Encoding and Decoding – Process Addressing – Failure Handling – Group Communication

**UNIT – III****(12 HOURS)**

Remote Procedure Calls : RPC models – Transparency of RPC–Stub generation–RPC messages– Marshaling arguments and results–Exception Handling–Lightweight RPC; Distributed Shared Memory: Introduction – General Architecture of DSM system – Design and Implementation Issues of DSM – Granularity – Structure of Shared Memory – Consistency Models – Replacement Strategy – Thrashing.

**UNIT – IV****(12 HOURS)**

Synchronization: Introduction – Clock Synchronization – Event Ordering – Mutual Exclusion – Deadlock – Election Algorithm–Process Management: Introduction–Process Migration– Threads.

**UNIT – V****(12 HOURS)**

Distributed File System: Introduction – Desirable features – File Models – File Accessing Models – File Sharing Semantics – File Caching Schemes – File Replication – Fault Tolerance – Atomic Transactions – Design Principles.

**Mapping of COs to POs and PSOs**

Course Outcome	PO Addressed PO1 to PO7	Correlation Level L/M/H	PSO Addressed PSO1 to PSO7		Correlation Level L/ M/ H	Cognitive Level K <sub>1</sub> to K <sub>6</sub>
CO1	PO1	H	PSO1		H	K <sub>1</sub>
CO2	PO1	M	PSO1		M	K <sub>2</sub>
CO3	PO2	H	PSO2	PSO4	H   M	K <sub>3</sub>
CO4	PO4	H	PSO6		H	K <sub>4</sub>
CO5	PO6	M	PSO3	PSO5	H   M	K <sub>5</sub>

(L – Low, M – Medium, H – High; K<sub>1</sub> – Remember, K<sub>2</sub> – Understand, K<sub>3</sub> – Apply, K<sub>4</sub> – Analyze, K<sub>5</sub> – Evaluate, K<sub>6</sub> – Create)

**Text and Reference books**

1. Pradeep K Sinha, —Distributed Operating Systems – Concepts and Design||, PHI, 2016
2. Andrew S Tanenbaum , —Distributed Operating Systems||, First Edition, PHI.2017
3. Abraham Silberchalz Peter B. Galvin, G.Gagne, Operating Systems Concepts||,Ninth edition, Addison Wesley Publishing Co., 2018.
4. Coulouris George, Dollimore Jean, Blair Gordon–Distributed systems- concepts and design Pearson 2017.

## CORE - 6 ALGORITHM LAB (USE C++/JAVA)

[CLTP2004]

### Course Objectives:

Implement algorithms using popularly used strategies like greedy, divide and conquer, dynamic programming etc.

### Course Prerequisites:

1. Exposure to introductory course on programming languages
2. Some Basic Mathematics

### Course Outcomes (COs):

At the end of the Course, students will be able to:

**CO1:** Implement algorithms for solving real-time problems

### Course Outline:

#### (Model List only, please add more algorithms based programs)

1. Sort a given set of elements using the Quick sort method and determine the time required to sort the elements. Repeat the experiment for different values of n.
2. Sort a given set of elements using merge sort method and determine the time required to sort the elements. Repeat the experiment for different of values of n.
3. Write a program to obtain the topological ordering of vertices in a given digraph.
4. Implement travelling salesman problem.
5. Implement the knapsack problem (0/1).
6. Print all the nodes reachable from a given starting node in a digraph using BFS method.
7. Check whether a given graph is connected or not using DFS method.
8. Write a program to implement binary search using divide and conquer technique
9. Write a program to implement insertion sort using decrease and conquer technique
10. Find minimum cost spanning tree of a given undirected path using a Prim's algorithm.
11. From a given vertex in a weighted connected graph, find shortest paths to other vertices using Dijkstra's algorithm.

### Mapping of COs to POs and PSOs

Course Outcome	PO Addressed PO1 to PO7				Correlation Level/ M/H				PSO Addressed PSO1 to PSO7				Correlation Level /M/ H				Cognitive Level K <sub>1</sub> to K <sub>6</sub>
	PO1	PO2	PO3	PO4	L	H	M	M	PSO1	PSO3	PSO6	PSO6	M	H	M	M	
CO1	PO1	PO2	PO3	PO4	L	H	M	M	PSO1	PSO3	PSO6	PSO6	M	H	M	M	K <sub>4</sub> , K <sub>5</sub> , K <sub>6</sub>

(L – Low, M – Medium, H – High; K<sub>1</sub> – Remember, K<sub>2</sub> – Understand, K<sub>3</sub> – Apply, K<sub>4</sub> – Analyze, K<sub>5</sub> – Evaluate, K<sub>6</sub> – Create)

**Course Objectives:**

1. To design stand-alone desktop-oriented GUI based Java applications using Swing and access the database using JDBC.
2. To understand the networking components to transfer data over networks.
3. To develop web application based on Java uses Servlet, JSP
4. To design applications using pre-built frameworks.

**Course Prerequisites:**

1. Experience in any of the Object Oriented Programming language
2. Basic understanding of the Java language
3. Basic knowledge on SQL

**Course Outcomes\* (COs):**

At the end of the Course, the student will be able to

- CO1:** Understand swing components and its usage.
- CO2:** Implement Networking and Data base connectivity in Java for given application.
- CO3:** Implement webpage with dynamic content and server side web application using Servlet and JSP.
- CO4:** Develop Java application using spring framework.

**Course Outline:**

**(This is a model List only, please add more programs)**

1. Write a Program in Java to implement Calculator using Swing technology
2. Write a Program that displays two textboxes for entering a students' Roll-no and Name with appropriate labels and buttons
3. Write a Java program that makes a connection with database using JDBC and prints metadata of this connection
4. Include the database connectivity to insert, update, delete and display of student information and display it
5. Write a java program for one way TCP communication for server and client, where server will response to client with current data and time.
6. Write a java program for two way TCP communication for server and client. It should look like a simple chat application
7. Write a java program for UDP Communication where client will send name of country and server will return the capital of that country

8. Create a simple calculator application that demonstrates the use of RMI. You are not required to create GUI.
9. Create Servlet That Prints Hello World.
10. Create Servlet That Prints Today's Date
11. Create Servlet for login page, if the username and password is correct then prints message "Hello username" else a message "login failed".
12. Create Servlet that uses cookies to store the number of times a user has visited the servlet
13. Create a Servlet for demo of KBC game. There will be continuous two or three pages with different MCQs. Each correct answer carries Rs. 10000. At the end as per user's selection of answers total prize he won should be declared. User should not be allowed to backtrack.
14. Create a Servlet that implements ServletContextAttributeListener interface such that a message dialog is displayed whenever an attribute is added or removed or replaced.
15. Create a Servlet filter that calculates server's response time and add it to response when giving it back to client.
16. Create a jsp that prints hello world.
17. Create jsp that prints current date and time.
18. Create a jsp that add and subtract two numbers.
19. Create a jsp for login module.
20. Create a web page that prints 1 to 10 using JSTL.
21. Create a custom JSP tag that prints current date and time. Use this tag into JSP page.

#### Mapping of COs to POs and PSOs

Course Outcome	PO Addressed	Correlation Level	PSO Addressed		Correlation Level		Cognitive Level
			PSO1	PSO4	M	H	
CO1	PO2	H	PSO1	PSO4	M	H	K <sub>1</sub>
CO2	PO3	H	PSO2		H		K <sub>6</sub>
CO3	PO3	H	PSO4		H		K <sub>6</sub>
CO4	PO2	H	PSO2		H		K <sub>3</sub>

(L – Low, M – Medium, H – High; K<sub>1</sub> – Remember, K<sub>2</sub> – Understand, K<sub>3</sub> – Apply, K<sub>4</sub> – Analyze, K<sub>5</sub> – Evaluate, K<sub>6</sub> – Create)

### Semester II

Course No	Course Type	Course Name	Contact Hrs./ Week	Credits
8	Core- 8	Advanced Web Technology	5	4
9	Core- 9	Machine Learning	5	4
10	Core- 10	Advanced DBMS	4	4
11	Core- 11	Software Project Management	4	4
12	Elective– 1 ( Select any one )	1.Free open source Software 2.Data Mining 3. Software Quality Assurance	4	3
13	Core - 12 Practical - 3	Advanced Web Technology Lab	4	2
14	Core - 13 Practical - 4	Machine Learning Lab using Python	4	2
<b>Subtotal</b>			<b>30</b>	<b>23</b>

## Core – 8 ADVANCED WEB TECHNOLOGY [C L T P 4 4 1 0]

### **Course Objectives:**

- Explore the backbone of webpage creation by developing .NET skill.
- Enrich knowledge about HTML control and web control classes
- Provide in-depth knowledge about ADO.NET
- Understand the need of usability, evaluation methods for web services

### **Course Outcomes**

- CO1:** Design a webpage with Web form fundamentals and web control classes
- CO2:** Recognize the importance of validation control, cookies and session
- CO3:** Apply the knowledge of ASP.NET object, ADO.NET data access and SQL to develop a client server model.
- CO4:** Recognize the difference between Data list and Data grid controls in accessing data

**(Total 60 HOURS)**

### **UNIT1: OVERVIEW**

**(12 HOURS)**

**OVERVIEW OF ASP.NET** - The .NET framework – Learning the .NET languages Data types – Declaring variables- Scope and Accessibility- Variable operations- Object Based manipulation- Conditional Structures- Loop Structures- Functions and Subroutines. Types, Objects and Namespaces: The Basics about Classes- Value types and Reference types - Advanced class programming-Understanding name spaces and assemblies. Setting Up ASP.NET and IIS

### **UNIT-II APPLICATIONS**

**(12 HOURS)**

Developing ASP.NET Applications - ASP.NET Applications: ASP.NET applications – Code behind-The Global.Asax application file – Understanding ASP.NET Classes-ASP.NET Configuration. Web Form fundamentals: A simple page applet-Improving the currency converter-HTML control classes-The page class – Accessing HTML server controls. Web controls: Web Control Classes – AutoPostBack and Web Control events –Accessing web controls. Using Visual Studio.NET: Starting a Visual Studio.NET Project - Web form Designer-Writing code - Visualstudio.NET debugging. Validation and Rich Controls: Validation – A simple Validation example – Understanding regular expressions - A validated customer forms. State management - Tracing, Logging, and Error Handling.

**UNIT–III WORKINGWITHDATA****(12 HOURS)**

Working with Data - Overview of ADO.NET - ADO.NET and data management- Characteristics of ADO.NET-ADO.NET object model. ADO.NET data access : SQL basics– Select , Update, Insert, Delete statements-Accessing data- Creating a connection-Using a command with a DataReader - Accessing Disconnected data - Selecting multiple tables – Updating Disconnected data. Data binding: Single value Data Binding- Repeated value data binding- Data binding with data bases. Datalist –Data grid–Repeater –Files, Streams and Email –Using XML

**UNIT-IV WEBSERVICES****(12 HOURS)**

Web Services - Web services Architecture: Internet programming then and now- WSDL–SOAP- Communicating with a web service-Web service discovery and UDDI. Creating Web services: Web service basics- The Stock Quote web service – Documenting the web service- Testing the web service – Web service Data types - ASP.NET intrinsic objects .Using web services: Consuming a web service- Using the proxy class-An example with Terra Service.

**UNIT–V ADVANCEDASP.NET****(12 HOURS)**

AdvancedASP.NET-Component Based Programming: Creating a simple component–Properties and state- Database components- Using COM components. Custom controls: User Controls- Deriving Custom controls. Caching and Performance Tuning: Designing and scalability–Profiling- Catching-Output catching- Data catching. Implementing security: Determining security requirements-TheASP.NET security model – Forms authentication-Windows authentication.

**CO mapped with PO and PSO**

Course Outcome	PO Addressed		Correlation Level		PSO Addressed		Correlation Level		Cognitive Level
CO1	PO1		H		PSO5		H		K <sub>1</sub> , K <sub>2</sub>
CO2	PO2	PO3	H	H	PSO5	PSO6	H	H	K6
CO3	PO2	PO3	H	H	PSO3	PSO4	H	H	K6
CO4	PO4	PO5	H	H	PSO4	PSO6	H	H	K <sub>3</sub>

(L – Low, M – Medium, H – High; K<sub>1</sub> – Remember, K<sub>2</sub> – Understand, K<sub>3</sub>– Apply, K<sub>4</sub>– Analyze, K<sub>5</sub>– Evaluate, K<sub>6</sub>– Create

**Text and Reference Books**

1. Mathew Mac Donald, “ASP.NET Complete Reference”, TMH2005.
2. Crouch Matt J, “ASP.NET and VB.NET Web Programming”, Addison Wesley2002.
3. J. Liberty, D.Hurwitz,“ProgrammingASP.NET”, ThirdEdition,O’REILLY,2006.

## Core – 9 MACHINE LEARNING

[C L T P 4 4 1 0]

### Course Objectives:

- To Learn about Machine Intelligence and Machine Learning applications
- To implement and apply machine learning algorithms to real- world applications.
- To identify and apply the appropriate machine learning technique to classification, pattern recognition, optimization and decision problems.
- To understand how to perform evaluation of learning algorithms and model selection.

### Course Outcomes

- CO1:** Have a good understanding of the fundamental issues and challenges of machine learning: data, model selection, model complexity, etc.
- CO2:** Have an understanding of the strengths and weaknesses of many popular machine learning approaches.
- CO3:** Appreciate the underlying mathematical relationships within and across Machine Learning algorithms and the paradigms of supervised and un-supervised learning.
- CO4:** Have an understanding of the strengths and weaknesses of many popular machine learning approaches.
- CO5:** Be able to design and implement various machine learning algorithms in a range of real- world applications.

### Course Outline

**(Total 60 HOURS)**

#### UNIT -1 INTRODUCTION

(12 HOURS)

Learning – Types of Machine Learning – Supervised Learning – The Brain and the Neuron – Linear Discriminants – Perceptron – Linear Separability– Linear Regression.

#### UNIT - II LINEAR MODELS

(12 HOURS)

Multi-layer Perception – Going Forwards – Going Backwards: Back Propagation Error – Multi-layer Perception in Practice – Examples of using the MLP – Overview – Deriving Back-Propagation – Radial Basis Functions and Spines – Concepts – RBF Network – Curse of Dimensionality – Interpolations and Basis Functions – Support Vector Machines

#### UNIT– III TREE AND PROBABILISTIC MODELS

(12 HOURS)

Learning with Trees – Decision Trees – Constructing Decision Trees – Classification and Regression Trees – Ensemble Learning – Boosting – Bagging – Different ways to Combine Classifiers – Probability and Learning – Data into Probabilities – Basic Statistics – Gaussian Mixture Models – Nearest Neighbor Methods – Unsupervised Learning – K means Algorithms –

Vector Quantization – Self Organizing Feature Map

**UNIT- IV DIMENSIONALITY REDUCTION AND EVOLUTIONARY MODELS (12 HOURS)**

Dimensionality Reduction – Linear Discriminant Analysis – Principal Component Analysis – Factor Analysis – Independent Component Analysis – Locally Linear Embedding – Isomap – Evolutionary Learning – Genetic algorithms – Genetic Offspring: - Genetic Operators – Using Genetic Algorithms – Reinforcement Learning – Overview – Getting Lost Example – Markov Decision Process

**UNIT – V GRAPHICAL MODELS (12 HOURS)**

Markov Chain Monte Carlo Methods – Sampling – Proposal Distribution – Markov Chain Monte Carlo – Graphical Models – Bayesian Networks – Markov Random Fields – Hidden Markov Models – Tracking Methods

**Mapping of COs, POs and PSOs:**

Course Outcome	PO Addressed PO1 to PO7			Correlation Level L/M/H			PSO Addressed PSO1 to PSO7			Correlation Level L/ M/ H			Cognitive Level K <sub>1</sub> to K <sub>6</sub>
										L	M	H	
CO1	PO1			H			PSO1			H			K <sub>1</sub>
CO2	PO2			M			PSO1	PSO2		H	H		K <sub>2</sub>
CO3	PO3			H			PSO3	PSO4		H	H		K <sub>3</sub>
CO4	PO3	PO4	PO7	M	M	M	PSO5			H			K <sub>4</sub>
CO5	PO5	PO6	PO7	H	H	H	PSO2	PSO3	PSO5	M	M	M	K <sub>5</sub> , K <sub>6</sub>

(L – Low, M – Medium, H – High; K<sub>1</sub> – Remember, K<sub>2</sub> – Understand, K<sub>3</sub> – Apply, K<sub>4</sub> – Analyze, K<sub>5</sub> – Evaluate, K<sub>6</sub> – Create)

**Text and Reference books**

1. Stephen Marsland, —Machine Learning – An Algorithmic Perspective||, Second Edition, Chapman and Hall/CRC Machine Learning and Pattern Recognition Series, 2014.
2. Tom M Mitchell, —Machine Learning||, First Edition, McGraw Hill Education, 2013.

## Core – 10 ADVANCED DBMS

[CLTP4310]

### **Course Objectives:**

- Acquire broad understanding of database concepts and database management system software and Emerging Trends in it.
- Learn the method of handling distributed and object databases.

### **Course Outcomes**

- CO1:** Recognize the importance of Various Data models and Architecture
- CO2:** Analyze and Design the normalized database schema
- CO3:** Decide the database for his problem
- CO4:** Develop database solutions
- CO5:** Write database queries in SQL, PL SQL and NoSQL

### **Course Outline**

**(Total 60 HOURS)**

#### **UNIT -1**

**(12 HOURS)**

Database design and the ER model: Overview – The Entity-Relationship model – Constraints – Removing Redundant Attributes in Entity Sets – Entity Relationship Diagrams-Reduction to relational schemas – Entity Relationship Design Issues – Extended E-R Features. Relational Database Design: Features of good relational Design – Atomic Domains–1NF to 5NF – Denormalization.

#### **UNIT - II**

**(12 HOURS)**

Indexing and Hashing: Basic Concepts – Ordered Indices – B + Tree Index Files – B + Tree Extensions – Multiple Key Access – Static Hashing – Dynamic Hashing – Comparison of Ordered Indexing and Hashing – Bitmap Indices. Transactions: Transaction Concept – A simple Transaction model – Storage Structure – Transaction Atomicity and Durability – Transaction Isolation – Serializability.

#### **UNIT– III**

**(12 HOURS)**

Concurrency Control: Lock based Protocols – Deadlock Handling – Multiple Granularity – Timestamp Based Protocols – Validation Based Protocols – Multiversion Schemes – Snapshot Isolation – Insert Operations, Delete Operations and Predicate Reads. Recovery Systems: Failure

Classification – Storage – Recovery and Atomicity – Recovery Algorithm – Buffer Management.

**UNIT- IV**

(12 HOURS)

Database System Architecture: Centralized and Client Server Architectures – Server System Architectures – Parallel Systems – Distributed Systems. Parallel Databases: Introduction – I/O parallelism – Inter-query parallelism – Intra-query parallelism – Intra-operation parallelism – Interoperation parallelism – Query Optimization. Distributed Databases: Homogeneous and Heterogeneous Databases – Distributed Data Storage – Distributed Transactions.

**UNIT – V**

(12 HOURS)

Object-Based Databases: Complex Data Types, Structured Types and Inheritance, Table Inheritance, array and Multi-set, Object Identity and Reference Types, Object Oriented versus Object Relational. XML: Motivation-Structure of XML Data- XML document schema-Querying and transformation- Application Program Interfaces to XML –Storage of XML Data- XML Application.

**Mapping of COs to POs and PSOs**

Course Outcome	PO Addressed PO1 to PO7	Correlation Level L/M/H	PSO Addressed PSO1 to PSO7	Correlation Level L/ M/ H	Cognitive Level K <sub>1</sub> to K <sub>6</sub>
CO1	PO1	H	PSO1	M	K <sub>2</sub>
CO2	PO5	H	PSO1	H	K <sub>4</sub>
CO3	PO1	H	PSO2	M	K <sub>4</sub>
CO4	PO2	H	PSO1	H	K <sub>3</sub>
CO5	PO1	H	PSO1   PSO4	H   M	K <sub>2</sub>

(L – Low, M – Medium, H – High); K<sub>1</sub> – Remember, K<sub>2</sub> – Understand, K<sub>3</sub> – Apply, K<sub>4</sub> – Analyze, K<sub>5</sub> – Evaluate, K<sub>6</sub> – Create

**Text and Reference books**

1. Abraham Silberschatz, Henry F. Korth and S. Sudarshan, — “Database System Concepts”, Sixth Edition, McGraw Hill International Edition, 2011.
2. C.J.Date, — “Introduction to Database Systems”, 8th Edition, Pearson Education, 2003.
3. Kogent Learning Solutions "Database Management Systems applications" Dreamtech Press, 2014

## Core – 11 SOFTWARE PROJECT MANAGEMENT

[CLTP 4310]

### Course Objectives:

- To understand the roles of the project manager
- To understand the threats and opportunities in project management
- To gain Expertise in size, effort and cost estimation techniques
- To understand the techniques available to keep the project's aims and objectives, under control
- To understand how to approach non-technical problems
- To appreciate management issues like team structure, group dynamics

### Course Outcome:

- CO1:** Comprehend the roles of the project manager
- CO2:** Identify the threats and opportunities in project management.
- CO3:** Gain knowledge about size, effort and cost estimation techniques
- CO4:** Apply the techniques available to keep the project's aims and objectives, under control.
- CO5:** Analyze the different approaches of non-technical problems
- CO6:** Appreciate the management issues like team structure, group dynamics.

### Course Outline

(Total 60 Hours)

#### UNIT-1

(12 HOURS)

Introduction to Software Project Management: Introduction, Basic Definitions - Software vs Other Types of Project, Contract Management and Technical Project Management, Activities Covered by Software Project Management, Plans, Methods and Methodologies, Some Ways of Categorizing Software Projects, Project Charter, Stakeholders, Setting Objectives, The Business Case, Project Success and Failure, What is Management? Management Control, Project Management Life Cycle, Traditional versus Modern Project Management Practices. Project Evaluation and Programme Management: Introduction, Business Case, Project Portfolio Management, Evaluation of Individual Projects, Cost-benefit Evaluation Techniques, Risk Evaluation, Programme Management, Managing the Allocation of Resources within Programmes, Strategic Programme Management, Creating a Programme, Aids to Programme Management, Some Reservations about Programme Management, Benefits Management. An Overview of Project Planning: Introduction to Step Wise Project Planning – steps 0 to 10

**UNIT – II**

(12 HOURS)

Selection of an Appropriate Project Approach: Introduction, Choosing Methodologies and Technologies, Software Processes and Process Models, Choice of Process Models, Structure versus Speed of Delivery, The Waterfall Model, The Spiral Model, Software Prototyping, Other Ways of Categorizing Prototypes, Incremental Delivery, Atern/Dynamic Systems Development Method, Rapid Application Development, Agile Methods, Extreme Programming (XP), Scrum, Lean Software Development, Managing Iterative Processes, Selecting the Most Appropriate Process Model. Software Effort Estimation: Introduction, Where are the Estimates Done? Problems with Over- and Under-Estimates, The Basis for Software Estimating, Software Effort Estimation Techniques, Bottom- up Estimating, The Top-down Approach and Parametric Models, Expert Judgment, Estimating by Analogy, Albrecht Function Point Analysis, Function Points Mark II, COSMIC Full Function Points, COCOMO II: A Parametric Productivity Model, Cost Estimation, Staffing Pattern, Effect of Schedule Compression, Capers Jones Estimating Rules of Thumb.

**UNIT – III**

(12 HOURS)

Activity Planning: Introduction, Objectives of Activity Planning, When to Plan, Project Schedules, Projects and Activities, Sequencing and Scheduling Activities, Network Planning Models, Formulating a Network Model, Adding the Time Dimension, The Forward Pass, Backward Pass, Identifying the Critical Path, Activity Float, Shortening the Project Duration, Identifying Critical Activities, Activity-on-Arrow Networks. Risk Management: Introduction, Risk, Categories of Risk, Risk Management Approaches, A Framework for Dealing with Risk, Risk Identification, Risk Assessment, Risk Planning, Risk Management, Evaluating Risks to the Schedule, Boehm's Top 10 Risks and Counter Measures, Applying the PERT Technique, Monte Carlo Simulation, Critical Chain Concepts. Resource Allocation: Introduction, Nature of Resources, Identifying Resource Requirements, Scheduling Resources, Creating Critical Paths, Counting the Cost, Being Specific, Publishing the Resource Schedule, Cost Schedules, Scheduling Sequence.

**UNIT – IV**

(12 HOURS)

Monitoring and Control: Introduction, Creating the Framework, Collecting the Data, Review, Visualizing Progress, Cost Monitoring, Earned Value Analysis, Prioritizing Monitoring, Getting the Project Back to Target, Change Control, Software Configuration Management (SCM). Managing Contracts: Introduction, Types of Contract, Stages in Contract Placement, Typical Terms of a Contract, Contract Management, Acceptance. Managing People in Software Environments: Introduction, Understanding Behavior, Organizational Behavior: A Background, Selecting the Right Person for the Job, Instruction in the Best Methods, Motivation, The Oldham–Hackman Job Characteristics Model, Stress, Stress Management, Health and Safety, Some Ethical and Professional Concerns.

**UNIT – V**

(12 HOURS)

Working in Teams: Introduction, Becoming a Team, Decision Making, Organization and Team Structures, Coordination Dependencies, Dispersed and Virtual Teams, Communication Genres, Communication Plans, Leadership. Software Quality: Introduction, The Place of Software Quality in Project Planning, Importance of Software Quality, Defining Software Quality, Software Quality Models, ISO 9126, Product and Process Metrics, Product versus Process Quality Management, Quality Management Systems, Process Capability Models, Techniques to Help Enhance Software Quality, Testing, Software Reliability, Quality Plans. Project Closeout: Introduction, Reasons for Project Closure, Project Closure Process, Performing a Financial Closure, Project Closeout Report.

**Mapping of COs to POs and PSOs**

Course Outcome	PO Addressed PO1 to PO7			Correlation Level L/M/H			PSO Addressed PSO1 to PSO7				Correlation Level L/ M/ H			Cognitive Level K <sub>1</sub> to K <sub>6</sub>	
CO1	PO1			H			PSO1				H			K <sub>1</sub> ,K <sub>2</sub>	
CO2	PO2	PO3		M	M		PSO2				M			K <sub>3</sub>	
CO3	PO4	PO7		M	M		PSO2,	PSO3	PSO4		M	M	M	K <sub>4</sub> ,K <sub>5</sub>	
CO4	PO4	PO5	PO7	M	M	M	PSO2,	PSO3	PSO4		M	M	M	K <sub>4</sub> ,K <sub>5</sub>	
CO5	PO5	PO7		M	M		PSO4	PSO5	PSO6		H	H	H	K <sub>5</sub>	
CO6	PO2,	PO5	PO6	M	M	M	PSO4,	PSO5	PSO6	PSO7	M	M	M	M	K <sub>6</sub>

(L – Low, M – Medium, H – High; K<sub>1</sub> – Remember, K<sub>2</sub> – Understand, K<sub>3</sub> – Apply, K<sub>4</sub> – Analyze, K<sub>5</sub> – Evaluate, K<sub>6</sub> – Create)

**Text and Reference books**

1. Software Project Management, Bob Hughes, Mike Cotterell, Rajib Mall, Tata McGraw Hill, 6th Edition 2018
2. Project Management and Tools & Technologies – An overview, Shailesh Mehta, SPD First Edition 2017
3. Software Project Management , Walker Royce, Pearson Education, 2005

## Elective 1- (a) FREE OPEN SOURCE SOFTWARE

[C L T P 3 4 1 0]

### Course Objectives:

- To familiarize fundamentals of the shell programming, pipes, input and output redirection Control structures, arithmetic in shell interrupt processing, functions, debugging shell scripts.
- To impart fundamentals of file concepts kernel support for file, File structure related system calls (file API's).
- To teach principles of operating system including File handling utilities, Security by file permissions, Process utilities, Disk utilities, Networking Commands, Basic Linux commands, Scripts and filters.
- To know the basics of algorithmic problem solving
- To read and write simple Python programs. To develop Python programs with conditionals and loops.
- To define Python functions and call them.
- To use Python data structures -- lists, tuples, dictionaries

### Course Outcomes

- CO1** Ability to use various Linux commands that are used to manipulate system operations at admin level and a prerequisite to pursue job as a Network administrator.
- CO2** Ability to write Shell Programming using Linux commands.
- CO3** Ability to design and write application to manipulate internal kernel level Linux File System.
- CO4** Develop algorithmic solutions to simple computational problems Read, write, execute by hand simple Python programs.
- CO5** Structure simple Python programs for solving problems.
- CO6** Decompose a Python program into functions

### Course Outline

(Total 45 hours)

#### UNIT 1: INTRODUCTION TO LINUX AND LINUX UTILITIES

(9 hours)

A brief history of LINUX - architecture of LINUX - features of LINUX - introduction to vi editor – Basic Linux commands- File handling utilities - Security by file permissions - process utilities - disk utilities - networking commands -Text Processing utilities and backup utilities.

**UNIT - II INTRODUCTION TO SHELLS****(9 hours)**

Linux Session - Standard Streams- Redirection – Pipes - Tee Command - Command Execution – Command Line Editing - Quotes - Command Substitution - Job Control – Aliases - Variables - Predefined Variables – Options - Shell/Environment Customization - Filters: Filters and Pipes - Concatenating files - Display Beginning and End of files - Cut and Paste – Sorting - Translating Characters - Files with Duplicate Lines - Count Characters - Words or Lines - Comparing Files.

**UNIT III – ALGORITHMIC PROBLEM SOLVING IN PYTHON****(9 hours)**

Algorithms, building blocks of algorithms (statements, state, control flow, functions) - notation (pseudo code, flow chart, programming language), algorithmic problem solving - simple strategies for developing algorithms (iteration, recursion). Illustrative problems: find minimum in a list-insert a card in a list of sorted cards - guess an integer number in a range - Towers of Hanoi.

**UNIT IV- EXPRESSION, STATEMENTS AND CONTROL STRUCTURES****(9 hours)**

Python interpreter and interactive mode - values and types - int, float, boolean, string, and list; variables – expressions – statements - tuple assignment - precedence of operators - comments; modules and functions - function definition and use - flow of execution - parameters and arguments; Illustrative programs: exchange the values of two variables, circulate the values of n variables. Conditionals: Boolean values and operators - conditional (if), alternative (if-else), chained conditional (if-elif-else); Iteration: state, while, for, break, continue, pass.

**UNIT V- FUNCTIONS****(9 hours)**

Fruitful functions: return values – parameters - local and global scope - function composition - recursion; Strings: string slices - immutability - string functions and methods - string module - Lists as arrays - Illustrative programs: square root, gcd, and exponentiation, sum an array of numbers, linear search, binary search. Files, Types of Files, Creating and Reading Text Data, File Methods to Read and Write Data, Reading and Writing Binary Files, The Pickle Module, Reading and Writing CSV Files, Python os and os.path Modules.

**Mapping of COs to POs and PSOs**

Course Outcome	PO Addressed PO1 to PO7			Correlation Level L/M/H			PSO Addressed PSO1 to PSO7			Correlation Level L/ M/ H			Cognitive Level K <sub>1</sub> to K <sub>6</sub>		
CO1	PO1			H			PSO1			H			K <sub>1</sub> ,K <sub>2</sub>		
CO2	PO2	PO3		M	M		PSO2			M			K <sub>3</sub>		
CO3	PO4	PO7		M	M		PSO2,	PSO3	PSO4		M	M	M	K <sub>4</sub> ,K <sub>5</sub>	
CO4	PO4	PO5	PO7	M	M	M	PSO2,	PSO3	PSO4		M	M	M	K <sub>4</sub> ,K <sub>5</sub>	
CO5	PO5	PO7		M	M		PSO4	PSO5	PSO6		H	H	H	K <sub>5</sub>	
CO6	PO2,	PO5	PO6	M	M	M	PSO4,	PSO5	PSO6	PSO7	M	M	M	M	K <sub>6</sub>

(L – Low, M – Medium, H – High; K<sub>1</sub> – Remember, K<sub>2</sub> – Understand, K<sub>3</sub>– Apply, K<sub>4</sub>– Analyze, K<sub>5</sub>–Evaluate, K<sub>6</sub>– Create)

### ***Text and Reference books***

- W. Richard. Stevens (2005), Advanced Programming in the UNIX Environment, 3rd edition, Pearson Education, New Delhi, India.
- Unix and shell Programming Behrouz A. Forouzan, Richard F. Gilberg.Thomson
- Allen B. Downey, “Think Python: How to Think like a Computer Scientist”, 2nd edition, Updated for Python 3, Shroff/O’Reilly Publishers, 2016.
- Guido van Rossum and Fred L. Drake Jr, —An Introduction to Python – Revised and updated for Python 3.2, Network Theory Ltd., 2011.
- Charles Dier bach, —Introduction to Computer Science using Python: A Computational Problem-Solving Focus, Wiley India Edition, 2013.
- Gowri shankar S, Veena A, “Introduction to Python Programming”, 1st Edition, CRC Press/Taylor & Francis, 2018. ISBN-13: 978-0815394372

## Elective 1- (b) DATA MINING

[CLTP3410]

### Course Objectives:

- Examine the types of the data to be mined.
- Explore and understand data mining algorithms.

### Course Outcomes:

- CO1:** To evaluate various mining techniques on complex data objects
- CO2:** To develop applications using Data Mining Tools.
- CO3:** To develop ability to design various algorithms based on data mining tools.
- CO4:** To develop further interest in research and design of new Data Mining techniques

### Course Outline

(Total 45 hours)

#### UNIT-1

(9 hours)

Data Mining and Data Preprocessing: Data Mining – Motivation – Definition – Data Mining on what Kind of Data –Functionalities – Classification – Data Mining Task Primitives – Major Issues in Data Mining .Data Preprocessing – Definition – Data Cleaning – Integration - Transformation – Data Reduction.

#### UNIT – II

(9 hours)

Data Warehousing: Definition -Data Warehouse Architecture- Multidimensional Data Model. Frequent Patterns, Associations: Market basket analysis - Association Rule, Support and Confidence - apriori algorithm - Generating association rule from frequent item set - Mining frequent item sets without candidate generation (FP- growth) - Overview of multilevel association rule - Multidimensional association rule- - closed item set - maximal item set.

#### UNIT – III

(9 hours)

Definition of Classification and Prediction – Classification by Decision Tree Induction - Bayesian Classification – Rule Based Classification – Classification by Back Propagation – Lazy Learners – K-Nearest Neighbor – Other Classification Methods.

#### UNIT – IV

(9 hours)

Cluster Analysis: Definition – Types of data in Cluster Analysis – Categorization of major Clustering Techniques – Partitioning Methods – Hierarchical Clustering – BIRCH - ROCK – Grid Based Methods – Model Based Clustering Methods – Outlier Analysis.

#### UNIT – V

(9 hours)

Spatial, Multimedia, Text and Web Data: Spatial Data Mining – Multimedia Data Mining – Text Mining – Mining the World Wide Web – Data Mining Applications – Trends in Data Mining. Data mining tool – Orange Tool.

## Mapping of COs to POs and PSOs

Course Outcome	PO Addressed PO1 to PO7			Correlation Level L/M/H			PSO Addressed PSO1 to PSO7			Correlation Level L/ M/ H			Cognitive Level K <sub>1</sub> to K <sub>6</sub>
CO1	PO1			H			PSO1			H			K <sub>1</sub> ,K <sub>2</sub>
CO2	PO2	PO3		M	M		PSO2			M			K <sub>3</sub>
CO3	PO4	PO7		M	M		PSO2,	PSO3	PSO4	M	M	M	K <sub>4</sub> ,K <sub>5</sub>
CO4	PO4	PO5	PO7	M	M	M	PSO2,	PSO3	PSO4	M	M	M	K <sub>4</sub> ,K <sub>5</sub>

(L – Low, M – Medium, H – High; K<sub>1</sub> – Remember, K<sub>2</sub> – Understand, K<sub>3</sub>– Apply, K<sub>4</sub>– Analyze, K<sub>5</sub>–Evaluate, K<sub>6</sub>– Create)

### Text and Reference books

1. Jiawei Han and Micheline Kambar, — “Data Mining Concepts and Technique:”, Second Edition, Elsevier, Reprinted 2008.
2. Marget H. Dunham, — “Data Mining Introductory and Advanced Concepts” Pearson Education 2003.
3. Pang-Ning Tan, Michael Steinbach and Vipin Kumar, - “Introduction to Data Mining”, Pearson Education, 2007.
4. G.K. Gupta, - “Introduction to Data Mining with Case Studies”, 3rd Edition, PHI, 2015.
5. <http://www.celta.paris-sorbonne.fr/anasem/papers/miscelanea/InteractiveDataMining.pdf>

## Elective 1- (c) SOFTWARE QUALITY ASSURANCE [C L T P 3 3 1 0]

### Course Objectives:

- To understand the basics of software quality
- To learn various metrics of software quality
- To introduce concepts behind designing of test cases
- To learn the procedure of debugging a given software

### Course Outcomes:

- CO1:** Analyze software documentations using inspections and walkthrough
- CO2:** Able to assess the quality and make improvements in the quality of software
- CO3:** Associate various software metrics in the right context
- CO4:** Able to List the components of software test plan
- CO5:** Able to handle automation testing tools for software testing

### Course Outline

**(Total 45 Hrs)**

#### UNIT-1

**(9 Hrs)**

INTRODUCTION - Introduction to Software Quality - Challenges – Objectives – Quality Factors – Components of SQA – Contract Review – Development and Quality Plans – SQA Components in Project Life Cycle – SQA Defect Removal Policies – Reviews.

#### UNIT – II

**(9 Hrs)**

TESTING METHODOLOGIES - Basics of Software Testing – Test Generation from Requirements – Finite State Models – Combinatorial Designs - Test Selection, Minimization and Prioritization for Regression Testing – Test Adequacy, Assessment and Enhancement.

#### UNIT – III

**(9 Hrs)**

TEST STRATEGIES - Testing Strategies – White Box and Black Box Approach – Integration Testing – System and Acceptance Testing – Performance Testing – Regression Testing - Internationalization Testing – Adhoc Testing – Website Testing – Usability Testing – Accessibility Testing.

#### UNIT – IV

**(9 Hrs)**

TEST AUTOMATION AND MANAGEMENT - Test plan – Management – Execution and Reporting – Software Test Automation – Automated Testing tools - Hierarchical Models of Software Quality – Configuration Management – Documentation Control.

**UNIT – V****(9 Hrs)**

SQA IN PROJECT MANAGEMENT Project progress control – costs – quality management standards – project process standards – management and its role in SQA – SQA unit.

**Mapping of COs to POs and PSOs**

Course Outcome	PO Addressed PO1 to PO7		Correlation Level L/M/H		PSO Addressed PSO1 to PSO7			Correlation Level L/ M/ H			Cognitive Level K <sub>1</sub> to K <sub>6</sub>
CO1	PO1		H		PSO1			H			K <sub>1</sub> ,K <sub>2</sub>
CO2	PO2	PO3	M	M	PSO2			M			K <sub>3</sub>
CO3	PO4	PO7	M	M	PSO2,	PSO3	PSO4	M	M	M	K <sub>4</sub> ,K <sub>5</sub>
CO4	PO4	PO5   PO7	M	M   M	PSO2,	PSO3	PSO4	M	M	M	K <sub>4</sub> ,K <sub>5</sub>
CO5	PO5	PO7	M	M	PSO4	PSO5	PSO6	H	H	H	K <sub>5</sub>

(L – Low, M – Medium, H – High; K<sub>1</sub> – Remember, K<sub>2</sub> – Understand, K<sub>3</sub>– Apply, K<sub>4</sub>– Analyze, K<sub>5</sub>– Evaluate, K<sub>6</sub>– Create)

**Text and References books**

1. Aditya Mathur, —Foundations of Software Testing, Pearson Education, 2008
2. Alan C Gillies, —Software Quality Theory and Management, Cengage Learning, Second Edition, 2003.
3. Daniel Galin, —Software Quality Assurance – from Theory to Implementation, Pearson Education, 2009
4. Robert Furtell, Donald Shafer, and Linda Shafer, "Quality Software Project Management", Pearson Education Asia, 2002.
5. Ron Patton, —Software Testing, Second Edition, Pearson Education, 2007
6. Srinivasan Desikan, Gopaldaswamy Ramesh, —Software Testing – Principles and Practices, Pearson Education, 2006
7. Yogesh Singh, "Software Testing", Cambridge University Press, 2012

***Course Objectives:***

- Explore the backbone of webpage creation by developing .NET skill.
- Enrich knowledge about HTML control and web control classes
- Provide in-depth knowledge about ADO.NET
- Understand the need of usability, evaluation methods for web services

***Course Outcomes***

- CO1:** Design a webpage with Web form fundamentals and web control classes
- CO2:** Recognize the importance of validation control, cookies and session
- CO3:** Apply the knowledge of ASP.NET object, ADO.NET data access and SQL to develop a client server model.
- CO4:** Recognize the difference between Data list and Data grid controls in accessing data

**List of Exercises Recommended**

**(The following are the minimum set of exercises suggested. The course coordinator may include more number of additional exercises in line with the course outcome)**

1. Implementing Currency Converter logic in ASP.NET
2. Demonstrate Page Event Tracking in ASP.NET
3. Program to generate online greeting card by using file uploading control
4. Including Advertisement Banners in a website.
5. Program to create a registration page using validation controls
6. Program using Session State variables and set different session timeout
7. Program using Query String in ASP.NET.
8. ASP.NET program to create, read and remove Cookies
9. Program to handle Data Grid and Data List controls in a web page
10. ASP.NET program to search and store student data.
11. Creating a simple web service in ASP.NET
12. Creating User control in ASP.NET and use it in simple web application.
13. Creating a Custom in ASP.NET and use it in simple web application.
14. Implementing web security using forms authentication in ASP.NET

**CO mapped with PO and PSO**

<b>Course Outcome</b>	<b>PO Addressed</b>		<b>Correlation Level</b>		<b>PSO Addressed</b>		<b>Correlation Level</b>		<b>Cognitive Level</b>
CO1	PO1		H		PSO5		H		K <sub>1</sub> , K <sub>2</sub>
CO2	PO2	PO3	H	H	PSO5	PSO6	H	H	K6
CO3	PO2	PO3	H	H	PSO3	PSO4	H	H	K6
CO4	PO4	PO5	H	H	PSO4	PSO6	H	H	K <sub>3</sub>

(L – Low, M – Medium, H – High; K<sub>1</sub> – Remember, K<sub>2</sub> – Understand, K<sub>3</sub> – Apply, K<sub>4</sub> – Analyze, K<sub>5</sub> – Evaluate, K<sub>6</sub> – Create)

## **Core – 13 (Practical) MACHINE LEARNING LAB USING PYTHON [C L T P 2004]**

### **Course Objectives:**

- To apply the concepts of Machine Learning to solve real-world problems
- To implement basic algorithms in clustering & classification applied to text & numeric data
- To implement algorithms emphasizing the importance of bagging & boosting in classification & regression
- To implement algorithms related to dimensionality reduction
- To apply machine learning algorithms for Natural Language Processing applications

### **Course Outcomes:**

- CO1:** To learn to use Weka tool for implementing machine learning algorithms related to numeric data
- CO2:** To learn the application of machine learning algorithms for text data
- CO3:** To use dimensionality reduction algorithms for image processing applications
- CO4:** To apply CRFs in text processing applications
- CO5:** To use fundamental and advanced neural network algorithms for solving real-world data

### **LIST OF EXERCISES RECOMMENDED:**

1. Solving Regression & Classification using Decision Trees
2. Root Node Attribute Selection for Decision Trees using Information Gain
3. Bayesian Inference in Gene Expression Analysis
4. Pattern Recognition Application using Bayesian Inference
5. Bagging in Classification
6. Bagging, Boosting applications using Regression Trees
7. Data & Text Classification using Neural Networks
8. Using Weka tool for SVM classification for chosen domain application
9. Data & Text Clustering using K-means algorithm
10. Data & Text Clustering using Gaussian Mixture Models
11. Dimensionality Reduction Algorithms in Image Processing applications
12. Application of CRFs in Natural Language Processing

**Mapping of COs, POs and PSOs:**

Course Outcome	PO Addressed PO1 to PO7			Correlation Level L/M/H			PSO Addressed PSO1 to PSO7			Correlation Level L/M/H			Cognitive Level K <sub>1</sub> to K <sub>6</sub>
										L	M	H	
CO1	PO1			H			PSO1			H			K <sub>1</sub>
CO2	PO2			M			PSO1	PSO2		H	H		K <sub>2</sub>
CO3	PO3			H			PSO3	PSO4		H	H		K <sub>3</sub>
CO4	PO3	PO4	PO7	M	M	M	PSO5			H			K <sub>4</sub>
CO5	PO5	PO6	PO7	H	H	H	PSO2	PSO3	PSO5	M	M	M	K <sub>5</sub> , K <sub>6</sub>

(L – Low, M – Medium, H – High; K<sub>1</sub> – Remember, K<sub>2</sub> – Understand, K<sub>3</sub> – Apply, K<sub>4</sub> – Analyze, K<sub>5</sub> – Evaluate, K<sub>6</sub> – Create)

### Semester -III

Semester ( 1)	Course No. ( 2)	Course Type ( 3)	Course Name (4)	Contact Hrs./ Week (5)	Credits (6)
III	15	Core-14	Software Testing	4	4
	16	Core-15	Soft Computing	4	4
	17	Core-16	Advanced Computer Networks	4	4
	18	Core-17	Research Methodology	4	4
	19	Elective - 2 ( Select any one )	<ol style="list-style-type: none"> <li>1. Cloud Computing</li> <li>2. Mobile Application Development</li> <li>3. Object Oriented Software Engineering</li> </ol>	4	3
	20	Core - 18 Practical - 5	Software Testing Lab	4	2
	21	Core –19	Mini Project	6	6

## Core –14 SOFTWARE TESTING

[CLTP 4 3 1 0]

### Course Objectives:

- To enable a clear understanding about software tester
- To apply software testing knowledge and engineering concepts to detect errors in the software
- To practice software oriented testing projects
- To prepare software testing techniques and tools for industry standards.

### Course Outcomes:

- Get an insight into the process of various software testing techniques
- Able to measure the performance of the using various metrics
- Able to evaluate the system with various testing techniques and strategies

### Course Outcomes:

- CO1:** Review the fundamental concepts of software testing and apply them
- CO2:** Evaluate the software with various testing strategies.
- CO3:** Interpret the testing results and decide upon the debugging strategies
- CO4:** Able to measure the performance of the software using various metrics
- CO5:** Test and Evaluate Object Oriented Software system

### Course Outline:

**(Total 60 hours)**

#### UNIT-1 SOFTWARE QUALITY ASSURANCE

**(12 Hrs)**

Introduction to Software Quality Engineering: What is software quality – Benefits of software quality – Software development life cycle model – Types of defects – Definitions used in software quality engineering - Software Quality Assurance and Quality Control -Software Configuration Management (SCM). Software Quality Assurance: Benefits of SQA – Role of SQA – SQA people – SQA plan –What is process– Process frame works. Reviews, Inspections and walk through: Management and Technical reviews -Inspection sand walk through–Inspection forms and check lists– Rate of Inspection–Inspection metrics-Estimating total number of defects in the software.

#### UNIT – II TESTING TECHNIQUES

**(12 Hrs)**

Introduction to Testing : Guiding Principles of testing– Composition of testing team–Essential skills of a tester–Types of Testing–Evaluating the quality of test cases–Techniques for reducing number of test cases – Requirements for effective testing – Test Oracle – Economics of Software testing – Handling defects – Risk in software testing –Requirements traceability matrix. White box (Structural) Testing: Introduction to control flow graph – Control flow testing –Basis path testing–Linear Code Sequence And Jump(LCSAJ)coverage or JJ–path coverage – Loop testing –

Data flow testing – Slice-based testing – Pitfalls of white box testing– Tools for white box testing. Integration Testing: Types of Integration testing–Functional Decomposition based Integration– Call graph-based Integration –Path-based Integration –Smoke testing.

**UNIT – III FUNCTIONAL & NON-FUNCTIONAL TESTING (12 Hrs)**

Functional Testing: Logic-based Testing – State Transition Testing – Use Case-based Testing – Syntax Testing –Domain Testing –Petry Net-based testing –Tools used in Functional testing. Non-functional, Acceptance and Regression Testing: Non-functional Testing – Acceptance Testing-Regression Testing.

**UNIT – IV INCORPORATING SPECIALIZED TESTING TECHNIQUES (12 Hrs)**

Testing of OO Software and Agile Testing: Basics of OO system– Overview of UML diagram– OO Testing–Quality Metrics for OO Software–Agile Testing. Test Management: Activities in Test Management–Evaluation of Test Effectiveness–Release Management – Tools used in Test management. Cloud Testing: Introduction to Cloud computing– Cloud testing–Testing as a Service (TaaS).

**UNIT – V TEST AUTOMATION & QUALITY METRICS (12 Hrs)**

Test Automation: Advantages and disadvantages of test automation – Activities in test Automation- Test Automation Frame work – Tools for Test Automation – Script languages in Test Automation. Metrics for Software Quality: Categories of Software metrics – Metrics program – Types of Metrics– Some commonly used Software Metrics. Tools for Quality Improvement: Basic Quality Control Tool – Check sheet – Cause and effect Diagram – Pareto Diagram – Histogram – Scatter Plot – Run chart – Control Chart – Orthogonal defect Classification.

**Mapping of COs, POs and PSOs:**

Course Outcome	PO Addressed PO1 to PO 7			Correlation Level L/M/H			PSO Addressed PSO 1 to PSO7		Correlation Level L/ M/ H		Cognitive Level K1 to K6
	PO1	PO2	PO3	H	M	L	PSO1	PSO2	H	M	
CO1	PO1	PO2		H			PSO1		H		K1
CO2	PO3	PO4	PO5	M	M	M	PSO1	PSO2	H	H	K5
CO3	PO1	PO6		H			PSO3	PSO4	H	H	K3, K4, K6
CO4		PO4			M		PSO4		M		K5, K6
CO5		PO5			M		PSO4		H		K4, K5

(L – Low, M – Medium, H – High; K<sub>1</sub> – Remember, K<sub>2</sub> – Understand, K<sub>3</sub>– Apply, K<sub>4</sub>– Analyze, K<sub>5</sub>– Evaluate, K<sub>6</sub>– Create)

***Text books and References***

1. Anirban Basu, "Software Quality Assurance, Testing and Metrics", PHI, 2015.
2. Sandeep Desai, Abhishek Srivastava, "Software Testing A Practical Approach", PHI,2016.
3. Srinivasan Desikan, Gopaldaswamy Ramesh, "Software Testing Principles and practices",Pearson,2018.
4. Aditya P Mathur, "Foundations of Software Testing", Pearson, 2019

## Core - 15 SOFT COMPUTING

[CLTP 4310]

### Course Objectives:

- Develop the skills to gain a basic understanding of neural network theory and fuzzy logic theory.
- Introduce students to artificial neural networks and fuzzy theory from an engineering perspective.

### Course Outcomes:

At the end of this course, the students should be able to:

**CO1:** To understand the fundamental theory and concepts of neural networks, identify different neural network architectures, algorithms, applications and their limitations.

**CO2:** Comprehend the fuzzy logic and the concept of fuzziness involved in various systems and fuzzy set theory.

**CO3:** Understand the concept of soft fuzzy sets, knowledge representation using fuzzy rules, approximate reasoning, fuzzy inference systems, and fuzzy logic.

**CO4:** Understand appropriate learning rules for each of the architectures and learn several Genetic algorithms paradigms and its applications

### Course Outline:

**(Total 60 hours)**

#### UNIT-1 INTRODUCTION

*(12 Hours)*

Introduction: Soft Computing Constituents – Soft Computing Vs Hard Computing – Characteristics - Applications - Artificial Neural Network (ANN): Fundamental Concept – Application Scope - Basic Terminologies – Neural Network Architecture – Learning Process – Basic Models of ANN: McCulloch-Pitts Model – Hebb Network – Linear Separability.

#### UNIT – II SUPERVISED LEARNING NETWORKS

*(12 Hours)*

Supervised Learning Networks: Perceptron Networks – Adaline and Madaline Networks – Back Propagation Network – Radial Basis Function Network. Associative Memory Networks – BAM – Hopfield Network - Boltzmann Machine. Unsupervised Learning Networks: Kohonen Self Organizing Network – Counter Propagation Network – ART Network.

#### UNIT – III FUZZY SETS

*(12 Hours)*

Fuzzy Sets: Basic Concept – Crisp Set Vs Fuzzy Set - Operations on Fuzzy Set – Properties of Fuzzy Sets – Fuzzy Relations: Concept – Fuzzy Composition – Fuzzy Equivalence and Tolerance Relation

- Membership Functions: Features – Fuzzification – Methods of Membership value assignments
- De-fuzzification – Methods.

**UNIT – IV FUZZY CONCEPTS**

(12 Hours)

Fuzzy Arithmetic – Extension Principle – Fuzzy Measures – Fuzzy Rules and Fuzzy Reasoning: Fuzzy Propositions – Formation of Rules – Decomposition of Rules – Aggregation of Rules – Approximate Reasoning – Fuzzy Inference and Expert Systems – Fuzzy Decision Making – Fuzzy Logic Control Systems.

**UNIT – V GENETIC ALGORITHM**

(12 Hours)

Genetic Algorithm: Fundamental Concept – Basic Terminologies – Traditional Vs Genetic Algorithm - Elements of GA - Encoding - Fitness Function – Genetic Operators: Selection – Cross Over - Inversion and Deletion - Mutation – Simple and General GA – The Schema Theorem - Classification of Genetic Algorithm – Genetic Programming – Applications of GA.

**Mapping of COs, POs and PSOs:**

Course Outcome	PO Addressed PO1 to PO7			Correlation Level L/M/H			PSO Addressed PSO1 to PSO7		Correlation Level L/ M/ H		Cognitive Level K <sub>1</sub> to K <sub>6</sub>
	PO1	PO2	PO3	M	M	H	PSO1	PSO2	H	M	
CO1	PO1	PO2		H			PSO1		H		K1
CO2	PO3	PO4	PO5	M	M	H	PSO1	PSO2	H	H	K5
CO3	PO1	PO6		H	M		PSO3	PSO4	H	M	K3, K4, K6
CO4	PO1	PO5		M	L		PSO1	PSO3	H	M	K2, K3, K4

(L – Low, M – Medium, H – High; K<sub>1</sub> – Remember, K<sub>2</sub> – Understand, K<sub>3</sub> – Apply, K<sub>4</sub> – Analyze, K<sub>5</sub> – Evaluate, K<sub>6</sub> – Create)

**Text and Reference books**

1. S.N. Sivanandam, S.N. Deepa, “Principles of Soft Computing”, Wiley India, 2007.
2. S. Rajasekaran, G.A.V. Pai, “Neural Networks, Fuzzy Logic, Genetic Algorithms”, Prentice Hall India, 2004.

## Core - 16 ADVANCED COMPUTER NETWORKS

[CLTP 4310]

### Course Objectives:

- To study communication network protocols, different communication layer structure
- To learn security mechanism for data communication
- To learn network simulator.

### Course Outcomes:

After completion of this course, students will be able to

**CO1:** Understand fundamental underlying principles of computer networking

**CO2:** Understand details and functionality of layered network architecture.

**CO3:** Apply mathematical foundations to solve computational problems in computer networking

**CO4:** Analyze performance of various communication protocols.

**CO5:** Compare routing algorithms

**CO6:** Practice packet /file transmission between nodes.

### Course Outline:

**(Total 60 hours)**

#### UNIT-1

*(12 Hours)*

INTRODUCTION TO NETWORKS & COMMUNICATION MEDIA: Uses – Network hardware – Network software – Reference Models. Example Networks: Internet – X.25 -ATM - Transmission media – Wireless Transmission – Telephone system – ISDN, ATM communication – satellite communication.

#### UNIT – II

*(12 Hours)*

DATA TRANSFER & ACCESS PROTOCOLS: Error detection and correction methods – Elementary protocols – Sliding window protocols - IEEE 802.2 Logical Link Control – Bluetooth: architecture – protocol stack – radio layer – baseband layer – L2CAP layer – frame structure.

#### UNIT – III

*(12 Hours)*

NETWORK LAYER PROTOCOLS: Routing algorithms – Congestion control: Principles – policies – Congestion control in VC subnets – congestion control in datagram subnets - Network layer in Internet: Architecture – IP protocol- IP Address – IPv6

#### UNIT – IV

*(12 Hours)*

TRANSPORT PROTOCOLS: Transport service – Transport protocols – Transport protocols in Internet: TCP and UDP

#### UNIT – V

*(12 Hours)*

APPLICATION LAYER ISSUES: Domain Name System – Electronic mail - Network security- Cryptography. Network Simulator: Basics of Computer Network Simulation – Introduction to Network Simulator 2 (NS2) – Basic Architecture – Installation – Directories and Convention – Running NS2 Simulation – Simulation Examples.

### Mapping of COs to POs and PSOs

Course Outcome	PO Addressed PO1 to PO7			Correlation Level L/M/H			PSO Addressed PSO1 to PSO7		Correlation Level L/ M/ H		Cognitive Level K <sub>1</sub> to K <sub>6</sub>
CO1	PO1			H			PSO1	PSO3	H		K <sub>1</sub>
CO2	PO1			H			PSO1		M		K <sub>2</sub>
CO3	PO2	PO4	PO5	H	M	M	PSO2	PSO4	M	M	K <sub>3</sub>
CO4	PO4	PO5		H	M		PSO2		H		K <sub>4</sub> , K <sub>5</sub>
CO5	PO3	PO5		M	H		PSO4	PSO5	H	M	K <sub>6</sub>
CO6	PO6	PO7		M	M		PSO6	PSO7	M	M	K <sub>6</sub>

(L – Low, M – Medium, H – High; K<sub>1</sub> – Remember, K<sub>2</sub> – Understand, K<sub>3</sub> – Apply, K<sub>4</sub> – Analyze, K<sub>5</sub> – Evaluate, K<sub>6</sub> – Create)

### Text and Reference books

1. Andrew S.Tanenbaum, “Computer Networks”, PHI, 5 th Edition, 2013
2. Behrouz A. Forouzan, “Data communication and Networking”, Tata McGraw-Hill, 4th Edition, 2006
3. William Stallings, “Data and Computer Communication”, 7th Edition, Pearson Education, 2007
4. Teerawat Ussaruyakul, EkramHossain, Introduction to Network Simulator NS2, Springer, 2009

## Core – 17 RESEARCH METHODOLOGY

[CLTP 4310]

### **Course Objectives:**

- To understand the importance of Research Methodology
  - To perform exploratory data analysis
  - To apply the statistical testing to prove the hypothesis
  - To provide the inference using quantitative data analysis
  - To make use of computer aids to analyze the data, prepare reports and presentations
- Able to evaluate methodology of teaching

### **Course Outcomes:**

- CO1:** Ability to apply different research approaches and methodologies  
**CO2:** Develop data collection instrument according to the underlying theoretical framework.  
**CO3:** Analyze quantitative data and qualitative data using software packages  
**CO4:** Construct and document an appropriate research design  
**CO5:** Discuss limitations and potential contribution to theory and practice of research  
**CO6:** Effectively apply the appropriate computer tools in each stage of research  
**CO7:** Ability to perform ICT based Teaching Methods

### **Course Outline:**

**(Total 60 hours)**

#### **UNIT-1**

**(12 hours)**

INTRODUCTION OF RESEARCH AND FORMULATION Motivation and Objectives – Research methods vs Methodology. Types of research – Descriptive vs. Analytical, Applied vs. Fundamental, Quantitative vs. Qualitative, Conceptual vs. Empirical. Defining and formulating the research problem - Selecting the problem - Necessity of defining the problem - Importance of literature review in defining a problem – Literature review – Primary and secondary sources – Reviews, treatise, monographs, patents –Critical literature review . RESEARCH DESIGN AND METHODS Research design – Basic Principles- Need of research design — Features of good design – Important concepts relating to research design.

#### **UNIT – II**

**(12 hours)**

Observation and Facts, Laws and Theories, Prediction and explanation, Induction, Deduction, Development of Models - Developing a research plan - Exploration, Description, Diagnosis, Experimentation - Determining experimental and sample designs. DATA COLLECTION Execution of the research - Observation and Collection of data - Methods of data collection.

#### **UNIT – III**

**(12 hours)**

DATA ANALYSIS Quantitative Methods: Online Quantitative Design and Survey – Descriptive Measures – Probability – Random Variables and Distribution Functions – Discrete Probability Distributions – Continuous Probability Distribution – Sampling Distributions – Theory of Estimation – Hypothesis Testing – Correlation – Regression – Principles of Sample Survey – Types of Sampling – Design of Experiments – CRD-RBD-LSD-Factor Analysis – Cluster Analysis –

Discriminant Analysis – Multiple Regression and Correlation – Canonical Correlation – Application of Statistical Software Packages. REPORTING AND THESIS WRITING Reporting and thesis writing – Structure and components of scientific reports - Types of report – Technical reports and thesis – Significance – Different steps in the preparation – Layout, structure and Language of typical reports – Illustrations and tables - Bibliography, referencing and footnotes – Use of Oral presentation – Software Packages for thesis Preparation– Planning – Preparation – Practice – Making presentation – Use of visual aids - Importance of effective communication.

**UNIT – IV (12 hours)**

APPLICATION OF RESULTS AND ETHICS Application of results and ethics - Environmental impacts - Ethical issues - ethical committees - Commercialization – Copy right – royalty –Intellectual property rights and patent law – Trade Related aspects of Intellectual Property Rights – Reproduction of published material – Plagiarism – Application of Plagiarism detection tools - Citation and acknowledgment - Reproducibility and accountability.

**UNIT – V (12 hours)**

METHODOLOGY OF TEACHING Teaching – Objectives of Teaching, Phases of Teaching – Teaching Methods: Lecture Method, Discussion Method, Discovery Learning, Inquiry, Problem Solving Method, Project method, Seminar – Integrating ICT in Teaching: Individualised Instruction, Ways for Effective Presentation with PowerPoint – Documentation – Evaluation: Formative, Summative & Continuous and Comprehensive Evaluation – Later Adolescent Psychology: Meaning, Physical, Cognitive, Emotional, Social and Moral Development – Teaching Later Adolescents.

**Mapping of Cos to POs and PSOs**

Course Outcome	PO Addressed PO1 to PO7	Correlation Level L/M/H	PSO Addressed PSO1 to PSO7	Correlation Level L/ M/ H	Cognitive Level K <sub>1</sub> to K <sub>6</sub>
CO1	PO1	M	PSO1	M	K <sub>1</sub>
CO2	PO3	M	PSO2	L	K <sub>2</sub>
CO3	PO3	M	PSO1	H	K <sub>3</sub>
CO4	PO3	H	PSO4	M	K <sub>4</sub> , K <sub>5</sub>
CO5	PO6	M	PSO2	H	K <sub>6</sub>
CO6	PO5	M	PSO1   PSO2	M   M	K <sub>6</sub>
CO7	PO7	L	PSO7	L	K <sub>3</sub>

(L – Low, M – Medium, H – High; K<sub>1</sub> – Remember, K<sub>2</sub> – Understand, K<sub>3</sub>– Apply, K<sub>4</sub>– Analyze, K<sub>5</sub>– Evaluate, K<sub>6</sub>– Create)

**Text and Reference books**

1. C R Kothari, Paperback “Research Methodology: Methods and Techniques”, 2014
2. Modern Language Association Handbook, Eight Edition, 2016
3. R. Paneerselvam, “Research Methodology” 2nd Edition, PHI, 2014
4. John W Creswel, Research Design: Qualitative, Quantitative, and Mixed Methods

Approaches, 3rd Edition, 2014

5. S.C. Gupta & V.K. Kapoor, Fundamentals of Mathematical Statistics, Sultan Chand & Sons, New Delhi, 2014 Edition.
6. S.C. Gupta & V.K. Kapoor, Fundamentals of Applied Statistics, Sultan Chand & Sons. 2014 Edition.
7. Sampath.K, Panneerselvam.A&Santhanam.S (1984), Introduction to Educational Technology (2nd Revised Ed.) New Delhi: Sterling Publishers.
8. Sharma.S.R(2003).Effective Classroom teaching modern methods, tools & techniques, Jaipur: Mangal Deep.
9. Vedanayagam.E.G (1989). Teaching Technology for College Teachers, Newyark: Sterling Publishers.

## Elective - 2 (a) CLOUD COMPUTING

[CLTP 3310]

### Course Objectives:

- To understand the concept of cloud computing.
- To appreciate the evolution of cloud from the existing technologies.
- To have knowledge on the various issues in cloud computing.
- To be familiar with the lead players in cloud.
- To appreciate the emergence of cloud as the next generation computing paradigm.

### Course Outcomes:

At the end of the course, the student will be able to

- CO1** : Interpret the key dimensions of the challenges of Cloud Computing
- CO2** : Examine the economics, financial, and technological implications for selecting cloud computing for own organization
- CO3** : Assessing the technological, and organizational capacity of employer's for actively initiating and installing cloud-based applications
- CO4** : Evaluate own organizations' needs for capacity building and training in cloud computing-related IT areas
- CO5** : Illustrate Virtualization for Data-Centre Automation

### Course Outline:

(Total 45 hours)

#### UNIT-1 INTRODUCTION

(9 hours)

Introduction to Cloud Computing – Definition of Cloud – Evolution of Cloud Computing – Underlying Principles of Parallel and Distributed Computing – Cloud Characteristics – Elasticity in Cloud – On-demand Provisioning.

#### UNIT – II CLOUD ENABLING TECHNOLOGIES

(9 hours)

Service Oriented Architecture – REST and Systems of Systems – Web Services – Publish-Subscribe Model – Basics of Virtualization – Types of Virtualization – Implementation Levels of Virtualization – Virtualization Structures – Tools and Mechanisms – Virtualization of CPU – Memory – I/O Devices – Virtualization Support and Disaster Recovery.

#### UNIT – III CLOUD ARCHITECTURE, SERVICES AND STORAGE

(9 hours)

Layered Cloud Architecture Design – NIST Cloud Computing Reference Architecture – Public, Private and Hybrid Clouds – IaaS – PaaS – SaaS – Architectural Design Challenges – Cloud Storage – Storage-as-a-Service – Advantages of Cloud Storage – Cloud Storage Providers – S3.

#### UNIT – IV RESOURCE MANAGEMENT AND SECURITY IN CLOUD

(9 hours)

Inter Cloud Resource Management – Resource Provisioning and Resource Provisioning Methods – Global Exchange of Cloud Resources – Security Overview – Cloud Security Challenges – Software-as-a-Service Security – Security Governance – Virtual Machine Security – IAM – Security Standards.

**UNIT – V CLOUD TECHNOLOGIES AND ADVANCEMENTS****(9 hours)**

Hadoop – MapReduce – Virtual Box – Google App Engine – Programming Environment for Google App Engine – Open Stack – Federation in the Cloud – Four Levels of Federation – Federated Services and Applications – Future of Federation.

**Mapping of COs to POs and PSOs**

Course Outcome	PO Addressed PO1 to PO8		Correlation Level L/M/H		PSO Addressed PSO1 to PSO8		Correlation Level L/ M/ H		Cognitive Level K <sub>1</sub> to K <sub>6</sub>
CO1	PO1		H		PSO1		L		K <sub>1</sub> , K <sub>2</sub>
CO2	PO2		L		PSO4		M		K <sub>3</sub>
CO3	PO3	PO4	M	M	PSO4	PSO5	M	M	K <sub>3</sub> , K <sub>4</sub>
CO4	PO3		H		PSO4		M		K <sub>5</sub>
CO5	PO6		H		PSO6		H		K <sub>5</sub>

(L – Low, M – Medium, H – High); K<sub>1</sub> – Remember, K<sub>2</sub> – Understand, K<sub>3</sub> – Apply, K<sub>4</sub> – Analyze, K<sub>5</sub> – Evaluate, K<sub>6</sub> – Create

**Text and Reference books**

1. Kai Hwang, Geoffrey C. Fox, Jack G. Dongarra, “Distributed and Cloud Computing, From Parallel Processing to the Internet of Things”, Morgan Kaufmann Publishers, 2012.
2. Rittinghouse, John W., and James F. Ransome, —Cloud Computing: Implementation, Management and Security, CRC Press, 2017.
3. RajkumarBuyya, Christian Vecchiola, S. ThamaraiSelvi, —Mastering Cloud Computing, Tata Mcgraw Hill, 2013.
4. Toby Velte, Anthony Velte, Robert Elsenpeter, “Cloud Computing – A Practical Approach, Tata Mcgraw Hill, 2009.
5. George Reese, “Cloud Application Architectures: Building Applications and Infrastructure in the Cloud: Transactional Systems for EC2 and Beyond (Theory in Practice), O’Reilly, 2009.

## Elective - 2 (b) MOBILE APPLICATION DEVELOPMENT [C L T P 3 3 1 0]

### Course Objectives:

- To facilitates classroom and laboratory learning,
- To enable letting students in developing competence and confidence in android programming
- To understand the entire Android Apps Development Cycle,
- To enable the students to independently create Android Applications

### Course Outcome

*At the end of the course, the student will be able to*

- CO1:** Understand the need for app development
- CO2:** Recognize the methods of Mobile app requirements and develop apps
- CO3:** Able to Understand different mobile application paradigms
- CO4:** Apply different methods and strategies of Mobile application design
- CO5:** Define and identify infrastructure requirement for Mobile Applications
- CO6:** Ability to conceptualize new ideas and present them as intellectual property

### Course Outline

**(Total : 45 hours)**

#### UNIT-1

**(9 hours)**

Basic of Android Programming: Introduction to Android OS, Setting up the Android Application Development Environment, Creating, Testing and Debugging Applications, Android Stack, Android applications structure, Activity life cycle, Understanding implicit and explicit intents.

#### UNIT – II

**(9 hours)**

User Interface in Android: Adaptive and responsive user interfaces, User Input Controls, Menus, Screen Navigation, RecyclerView, Drawables, Themes and Styles, Fragments Fragment Life Cycle, Introduction to Material Design, Testing the user interface.

#### UNIT – III

**(9 hours)**

Background tasks: AsyncTask, AsyncTaskLoader, Connecting App to Internet, Broadcast receivers, Services, Notifications, Alarm managers. Sensor, Location and Maps: Sensor Basic, Motion and Position Sensors, Location services, Google maps API, Google Places API.

#### UNIT – IV

**(9 hours)**

Working with data in Android: Shared Preferences, App Setting, SQLite primer, Store data using SQLite database, Content Providers, Content Resolver, Loader

#### UNIT – V

**(9 hours)**

Performance Improvement of App: Performance Parameters, Profiling Tools, Rendering and Layout, Garbage Collection and Memory Leaks, Best Practices. Publishing Your App: Preparing for publishing, Signing and preparing the graphics, publishing to the Android Market

## Mapping of COs to POs and PSOs

Course Outcome	PO Addressed PO1 to PO7		Correlation Level L/M/H		PSO Addressed PSO1 to PSO7	Correlation Level L/ M/ H		Cognitive Level K <sub>1</sub> to K <sub>6</sub>
CO1	PO1		M		PSO2	M		K <sub>1</sub> , K <sub>2</sub>
CO2	PO1	PO2	M	H	PSO2	H		K <sub>1</sub>
CO3	PO3	PO4	M	L	PSO2	M		K <sub>2</sub>
CO4	PO3		H	M	PSO4   PSO2	M	L	K <sub>2</sub>
CO5	PO5		M		PSO4	M		K <sub>3</sub>
CO6	PO6	PO7	M	M	PSO5	M		K <sub>5</sub>

(L – Low, M – Medium, H – High); K<sub>1</sub> – Remember, K<sub>2</sub> – Understand, K<sub>3</sub> – Apply, K<sub>4</sub> – Analyze, K<sub>5</sub>–Evaluate, K<sub>6</sub>– Create

### ***Text books and References***

1. Android: A Programming Guide by J.F. DiMarzio
2. Hello, Android: Introducing Google's Mobile Development Platform by Ed Burnett
3. Programming android by ZigurdMednieks
4. Android User Interface Design: Turning Ideas and Sketches into Beautifully Designed Apps bylan G. Clifton
5. Android Developer Fundamental Course by Google.
6. Advance Android Developer Course by Google.

## **Elective - 2 (c) OBJECT ORIENTED SOFTWARE ENGINEERING [C L T P 3 3 10]**

### **Course Outcomes:**

At the end of the course, the student will be able to

**CO1:** Recognize the areas of problem solving that needs object oriented software development

**CO2:** Describe and develop various approaches for object oriented design

**CO3:** Able to choose and design suitable UML diagrams and methods

**CO4:** Able to apply correct testing methods and maintain software systems.

**CO5:** Able to estimate the object oriented application by applying metric data

### **Course Objectives:**

- To understand the phases in object oriented software development
- To gain fundamental concepts of requirements engineering and analysis.
- To know about the different approach for object oriented design and its methods
- To learn about how to perform object oriented testing and how to maintain software
- To provide various quality metrics and to ensure risk management.

### **Course Outline:**

**(Total: 45 hours)**

#### **UNIT-1**

**(9 hours)**

Introduction - Object-Orientation - Object-Oriented Methodologies - Object-Oriented modeling and terminologies - Conventional Software Life Cycle Models - Object-Oriented Software Life Cycle Models-Selection of Software Development Life Cycle Models.

#### **UNIT – II**

**(9 hours)**

Software Requirements - Requirement Elicitation Techniques - Initial requirements Document - Use Case Approach - Characteristics of a Good Requirement – Software Requirements Specification Document – Requirements Change Management – Object-Oriented Software Estimation – Lorenz and Kidd Estimation Method – Use Case Points Method – Class Point Method - Object-Oriented Function Point – Risk Management.

#### **UNIT – III**

**(9 hours)**

Object-Oriented Analysis – Identification of Classes - Identification of Relationships – Identifying State and Behavior – Moving towards Object-Oriented Design - Object-Oriented Design – Interaction Diagrams – Sequence Diagrams – Collaboration Diagrams – Refinement of Use Case Description – Refinement of Classes and Relationships – Identification of Operations to Reflect the Implementation Environment – Construction of Detailed Class Diagram – Generating Test Cases from Use Cases - Object-Oriented Design Principles for Improving Software Quality.

#### **UNIT – IV**

**(9 hours)**

Moving towards Implementation – Activity Diagrams – State chart Diagrams – Storing Persistent Data in Database – Implementing the Classes – Software Quality and Metrics – Software Quality Models – Measurement Basics – Analyzing the Metric Data – Metrics for Measuring Size and

Structure – Measuring Software Quality - Object-Oriented Metrics.

**UNIT – V**

**(9 hours)**

Software Testing – Software Verification Techniques – Checklist: A Popular Verification Tool – Functional Testing – Structural Testing – Class Testing – State- Based Testing – Mutation Testing – Levels of Testing – Software Testing Tools – Software Maintenance – Categories of Software Maintenance – Challenges of Software Maintenance – Maintenance of Object-Oriented Software - Software Rejuvenation - Estimation of Maintenance Effort – Configuration Management – Regression Testing.

**Mapping of COs to POs and PSOs**

<b>Course Outcome</b>	<b>PO Addressed</b> PO1 to PO7	<b>Correlation Level</b> L/M/H	<b>PSO Addressed</b> PSO1 to PSO7	<b>Correlation Level</b> L/ M/ H	<b>Cognitive Level</b> K <sub>1</sub> to K <sub>6</sub>
CO1	PO1	H	PSO1	H	K <sub>1</sub> ,
CO2	PO3	L	PSO4	M	K <sub>5</sub>
CO3	PO4	M	PSO1	M	K <sub>4</sub>
CO4	PO1	H	PSO1	H	K <sub>1</sub>
CO5	PO7	M	PSO7	M	K <sub>6</sub>

(L – Low, M – Medium, H – High); K<sub>1</sub> – Remember, K<sub>2</sub> – Understand, K<sub>3</sub> – Apply, K<sub>4</sub> – Analyze, K<sub>5</sub>–Evaluate, K<sub>6</sub>– Create

***Text and Reference books***

1. Yogesh Singh, Ruchika Malhotra, “Object – Oriented Software Engineering”, PHI Learning Private Limited, First edition, 2012.
2. Roger Y.lee, “Object-Oriented Software Engineering with UML: A Hands-on Approach”, Nova Science Publishers, Incorporated, 2019.
3. ZeynepAltan, “Applications and Approaches to Object-Oriented Software Design: Emerging Research and Opportunities”, Part of the Advances in Systems Analysis, Software Engineering, and High Performance Computing Book Series, 2020

## Core 18 - (Practical) Software Testing LAB

### Course objectives

- *Using language(C,C++,JAVA) to derive test cases for any given problem*
- *To apply the appropriate technique for the design of flow graph.*
- *To create appropriate document for the software artifact.*

### Course outcomes

At the end of the course the students will be able to

- CO1:** List out the requirements for the given problem Design and implement the solution for given problem in any programming
- CO2:** Analyze the requirements for the given problem statement Design and implement various solutions for the given problem
- CO3:** Employ various design strategies for problem solving.
- CO4:** Construct control flow graphs for the solution that is implemented
- CO5:** Create appropriate document for the software artifact

### LIST OF EXPERIMENTS

(Following is only a limited list of experiments. The course in charge may include other experiments related to the course.)

1. Design and develop a program in a language of your choice to solve the triangle problem defined as follows: Accept three integers which are supposed to be the three sides of a triangle and determine if the three values represent an equilateral triangle, isosceles triangle, scalene triangle, or they do not form a triangle at all. Assume that the upper limit for the size of any side is 10. Derive test cases for your program based on boundary-value analysis, execute the test cases and discuss the results.
2. Design, develop, code and run the program in any suitable language to solve the commission problem. Analyze it from the perspective of boundary value testing, derive different test cases, execute these test cases and discuss the test results.
3. Design, develop, code and run the program in any suitable language to implement the NextDate function. Analyze it from the perspective of boundary value testing, derive different test cases, execute these test cases and discuss the test results.
4. Design and develop a program in a language of your choice to solve the triangle problem defined as follows: Accept three integers which are supposed to be the three sides of a

triangle and determine if the three values represent an equilateral triangle, isosceles triangle, scalene triangle, or they do not form a triangle at all. Assume that the upper limit for the size of any side is 10. Derive test cases for your program based on equivalence class partitioning, execute the test cases and discuss the results.

5. Design, develop, code and run the program in any suitable language to solve the commission problem. Analyze it from the perspective of equivalence class testing, derive different test cases, execute these test cases and discuss the test results.
6. Design, develop, code and run the program in any suitable language to implement the NextDate function. Analyze it from the perspective of equivalence class value testing, derive different test cases, execute these test cases and discuss the test results.
7. Design and develop a program in a language of your choice to solve the triangle problem defined as follows: Accept three integers which are supposed to be the three sides of a triangle and determine if the three values represent an equilateral triangle, isosceles triangle, scalene triangle, or they do not form a triangle at all. Derive test cases for your program based on decision-table approach, execute the test cases and discuss the results
8. Design, develop, code and run the program in any suitable language to solve the commission problem. Analyze it from the perspective of decision table-based testing, derive different test cases, execute these test cases and discuss the test results
9. Design, develop, code and run the program in any suitable language to solve the commission problem. Analyze it from the perspective of dataflow testing, derive different test cases, execute these test cases and discuss the test results.
10. Design, develop, code and run the program in any suitable language to implement the binary search algorithm. Determine the basis paths and using them derive different test cases, execute these test cases and discuss the test results.
11. Design, develop, code and run the program in any suitable language to implement the quicksort algorithm. Determine the basis paths and using them derive different test cases, execute these test cases and discuss the test results.
12. Design, develop, code and run the program in any suitable language to implement an absolute letter grading procedure, making suitable assumptions. Determine the basis paths and using them derive different test cases, execute these test cases and discuss the test results.

### Mapping of COs to POs and PSOs

Course Outcome	PO Addressed PO1 to PO7		Correlation Level L/M/H		PSO Addressed PSO1 to PSO7	Correlation Level L/ M/ H	Cognitive Level K <sub>1</sub> to K <sub>6</sub>
CO1	PO1	PO4	H	H	PSO1	H	K <sub>1</sub> , K <sub>4</sub>
CO2	PO5		L		PSO2	H	K <sub>5</sub>
CO3	PO5		M		PSO5	H	K <sub>2</sub>
CO4	PO4		H		PSO6	M	K <sub>3</sub>
CO5	PO6		H		PSO6	M	K <sub>6</sub>

(L – Low, M – Medium, H – High); K<sub>1</sub> – Remember, K<sub>2</sub> – Understand, K<sub>3</sub> – Apply, K<sub>4</sub> – Analyze, K<sub>5</sub>–Evaluate, K<sub>6</sub>– Create

## Core – 19 Mini Project work

### GUIDELINES FOR MINI PROJECT WORK

#### Objective

The Mini project is aimed to enable the third semester student to study of Project development and permit them to design and develop a small scale solution to some real world problem in any one of the core area of study. Student may undertake the project in her/his own area of interest under the supervision of one of the faculty member, complete within the third semester and the whole mini project work is to be evaluated continuously by the internal examiner (Guide) and by the External examiner during the end semester examination. The mini project enables the student to get a prior exposure to carry out the Major project at Fourth Semester

**Mode of Mini Project :** Individual Project

**Nature of Mini Project:** Every student shall undertake a unique project title (Novel Concept/ idea/system or a small research problem, which shall be designed and implemented using available software development tool /programming language) approved by her/his guide.

**Guide:** Each Student shall be allotted under the Guidance of one Department faculty member by the Programme coordinator/Head

**Duration :** One semester - (6 hours per week)  
Student carryout the Mini Project work in her/his College itself. In case of Company project, students are permitted to do the Mini project work in reputed IT companies without affecting the minimum attendance and other classes of the third semester

**Continuous Assessment:** Based on periodic reviews (Three reviews during the Semester. Tentative review dates are decided by the department and to be intimated to the students at the beginning of the third Semester)

#### Evaluation criteria

*Each student is evaluated by the Internal Examiner (Guide) continuously during the respective semester. External Examination will be conducted at the end of the respective semester.*

Maximum Marks	Internal	External
100	50	50

**Passing Criteria:** Student shall secure a minimum of **50 % marks in the external** evaluation and shall secure a **minimum of 50 % marks in combined Internal and External evaluation**. (There is no passing minimum for the internal evaluation)

<b>Internal (CIA) (50 Marks)</b> (All the three reviews are mandatory)		<b>External (50 Marks)</b>	
<b>Review I</b> (Problem identification, Title & Abstract submission, Novelty of the idea, proposed outcomes, issues in existing methods, tools to be used)	15 Marks	Both Internal and External Examiner Shall evaluate the student based on the following criteria at the end of the semester: (Guide or any other department faculty decided by the HOD shall be internal examiner. External Examiner will be appointed by the COE)	
<b>Review II</b> System Design / Database Design or Research Methodology / Algorithms and Techniques/ detailed Implementation plan	15 Marks	<b>Internal Examiner</b>  Project Report	20 Marks
<b>Review III</b> System Implementation status, Testing, demo of working system and completion of report writing	20 Marks	<b>External Examiner shall evaluate under the following criteria</b> <ul style="list-style-type: none"> <li>• Presentation of the Mini Project</li> <li>• Demonstration of the Mini project working</li> <li>• Viva -voce</li> </ul>	10 Marks  10 Marks  10 Marks
<b>Total</b>	<b>50 Marks</b>		<b>50 Marks</b>

Semester IV				
Course NO	Course Type	Course Name	No of Hours per Week	Credits
22	Core – 20	Major Project	30	16
<b>Subtotal</b>			<b>30</b>	<b>16</b>

### Core - 20 Major Project work

#### GUIDELINES FOR MAJOR PROJECT WORK

##### Objective

The Major project is aimed to enable the Fourth semester student to design and develop a standard solution to one of the significant real world problem in any one of the core area of study. Student may undertake the major project in her/his own area of interest under the supervision of one of the faculty member, They may also be permitted to undertake the Major project in a reputed IT firm also with prior permission from the Department Head. The Major project shall be completed within the fourth semester and the whole Major project work shall be evaluated continuously by the internal examiner (Guide) and by the External examiner during the end semester examination. The Major project enables the student to get a prior exposure to project development that enhances their employability skills.

Maximum Marks :	Internal	External
100	50	50

**Mode of Major Project:** Individual Project

**Nature of Major Project:** Every student shall undertake a unique project title (Novel Concept/ idea/system or a small research problem, which shall be designed and implemented using available software development tool /programming language) approved by her/his guide.

**Guide:** Each Student shall be allotted under the Guidance of one Department faculty member by the Programme coordinator/Head

**Duration :** One semester - (30 hours per week) Major project students may also opt for company projects with prior permission from the Head of the Department/Principal

**Continuous Assessment:** Based on periodic reviews (Three reviews during the Semester. Tentative review dates are decided by the department and to be intimated to the students at the beginning of the fourth Semester)

### Evaluation criteria

*Each student is evaluated by the Internal Examiner (Guide) continuously during the respective semester. External Examination will be conducted at the end of the respective semester.*

**Passing Criteria:** Student shall secure a minimum of **50 % marks in the external** evaluation and shall secure a **minimum of 50 % marks in combined Internal and External evaluation**. (There is no passing minimum for the internal evaluation)

<b>Internal (50 Marks)</b> (All the three reviews are mandatory)		<b>External (50 Marks)</b>	
<b>Review I</b> (Problem identification, Title & Abstract submission, Novelty of the idea, proposed outcomes, issues in existing methods, tools to be used)	15 Marks	Both Internal and External Examiner Shall evaluate the student based on the following criteria at the end of the semester: (Guide or any other department faculty decided by the HOD shall be internal examiner. External Examiner will be appointed by the COE)	
<b>Review II</b> System Design / Database Design / Methodology / Algorithms and Techniques/ detailed Implementation plan	15 Marks	<b>Internal Examiner</b>  Project Report	20 Marks
<b>Review III</b> System Implementation status, Testing, outcomes and report writing	20 Marks	<b>External Examiner shall evaluate under the following criteria</b> <ul style="list-style-type: none"> <li>• Presentation of the Project</li> <li>• Demonstration of the working project</li> <li>• Viva -voce</li> </ul>	10 Marks  10 Marks  10 Marks
<b>Total</b>	<b>50 Marks</b>		<b>50 Marks</b>