LORDS INSTITUTE OF ENGINEERING AND TECHNOLOGY
(An Autonomous Institution)
DEPARTMENT OF INFORMATION TECHNOLOGY
SCHEME OF INSTRUCTIONS & EXAMINATIONS [LR-23]
(W.E.F Academic Year 2025-26)
B.E. V-Semester

S. No.	Course Code	Course Category	Course Title	Scheme of Instructions				Scheme of Examination			CREDITS
				L	T	P/ D	Contact Hours/Week	CIE	SEE	Duration in Hours	
Theory Course											
1	U23CS501	PCC	Automata Theory, Languages and Computation	3	1	-	4	40	60	3	4
2	U23IT501	PCC	Design and Analysis of Algorithms	3	-	-	3	40	60	3	3
3	U23CM502	PCC	Artificial Intelligence	3	-	-	3	40	60	3	3
4	U23IT502	PCC	Full Stack Development	3	-	-	3	40	60	3	3
5	--	OEC	Open Elective - I	3	-	-	3	40	60	3	3
Practical / Laboratory Course											
6	U23IT5L1	PCC	Full Stack Development Lab	-	-	3	3	25	50	3	1.5
7	U23CM5L2	PCC	Artificial Intelligence Lab	-	-	3	3	25	50	3	1.5
Project											
8	U23IT5P1	PROJ	Internship (During vacation period after 4Sem.)	-	-	2	2	50	-	-	1
Skill Development Course											
9	U23MA5L1	BSC	Aptitude and Reasoning	-	-	2	2	25	50	3	1
Total				**15**	**1**	**10**	**26**	**325**	**450**	-	**21**

L: Lecture*(Hrs/Wk/Sem)* **T**: Tutorial *(Hrs/Wk/Sem)* **P**: Practical / **D**: Drawing *(Hrs/Wk/Sem)*
CIE: Continuous Internal Evaluation **SEE**: Semester End Examination (Univ.Exam)
OEC: Open Elective Courses **PCC:** Programme Core Courses
BSC: Basic Science Courses **MA**: Mathematics
IT: Information Technology **PCC:** Programme Core Courses

Note:
1. Each contact hour is a Clock Hour.
2. The duration of the practical class is three hours, however it can be extended wherever necessary, to enable the student to complete the experiment.

Course Code	Course Title						Core/Elective
U23CS501	AUTOMATA THEORY, LANGUAGES AND COMPUTATION						CORE
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
Discrete Mathematics	L	T	D	P			
	3	1	-	-	40	60	4

Course Objectives

Develop ability to

1. Develop a formal notation for strings, languages and machines.
2. Understand Regular Expression and algebraic laws.
3. Design context free grammars and PDA, to generate strings from a context free language and Convert them into normal forms.
4. Identify the hierarchy of formal languages, grammars and machines.
5. Distinguish between computability and non-computability and Decidability and undecidability.

Course Outcomes

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

1. Gain knowledge of the various abstract machines.
2. Use regular languages and regular expression for constructing different finite state machines.
3. Understand and design different types of grammars.
4. Construct Push down Automata.
5. Construct Turing Machine.

UNIT-I

Introduction: Introduction to Finite Automata, Structural Representations, Basic Concepts of Automata Theory – Symbol, Alphabets, Strings, Languages, Kleene Closure, Positive Closure.
Finite Automata without Output: Deterministic Finite Automata, Nondeterministic Finite Automata, Finite Automata with Epsilon-Transitions, Conversion of NFA to DFA

UNIT-II

Regular Expressions: Finite Automata and Regular Expressions, Applications of Regular Expressions, Algebraic Laws for Regular Expressions, Properties of regular sets, Conversion of Finite Automata to Regular Expressions.

Pumping Lemma for Regular Languages: Pigeon Hole Principle, Statement of the pumping lemma, Applications of the Pumping Lemma.

Closure Properties of Regular Languages: Closure properties of Regular languages, Decision Properties of Regular Languages, Minimization of Automata- Equivalence, Myhill - Nerode Theorem

UNIT-III

Context-Free Grammars: Definition of Context-Free Grammars, Derivations Using a Grammar, Leftmost and Rightmost Derivations, the Language of a Grammar, Sentential Forms, Parse Trees, Ambiguity in Grammars.
Push down Automata: Definition of the Pushdown Automaton, the Languages of a PDA, Equivalence of PDA's and CFG's, Acceptance by final state & empty stack, Deterministic PDA Conversion of CFG to PDA, Conversion of PDA to CFG.

UNIT-IV

Normal Forms for Context- Free Grammars: Grammar Simplification, Chomsky Normal form, Greibach Normal form.

Context-Free Languages: Statement of pumping lemma, Applications pumping lemma

Properties of Context-Free Languages: Closure properties of CFL's, Decision Properties of CFL's.

UNIT-V

Turing Machines: Introduction, Techniques for construction of Turing machines, Modifications of TM, TM as enumerator, Restricted TM.

Undecidability: Recursive and Recursively enumerable languages, Halting problem, Rice Theorem, Post's correspondence problem, P, NP, NP-Complete, NP-Hard Problems, Chomsky's Hierarchy–Regular grammars, Unrestricted grammar, CSL.

Suggested Reading:

1. Introduction to Automata Theory, languages, and Computation, John E. Hopcroft, Rajeev Motwani and Jeffrey D. Ullman, Pearson Education India, 3rd Edition,2008.
2. Automata and Computability, Undergraduate Texts in Computer Science, Dexter C. Kozen, Springer, 2015.
3. Introduction to the Theory of Computation, Michael Sipser, PWS Publishing 3rd Edition,2014.
4. Introduction to Languages and the Theory of Computation, John Martin, Tata McGraw Hill, 3rd Edition, 2002.

Course Code	Course Title						Core/Elective
U23IT501/ U23IT602	DESIGN AND ANALYSIS OF ALGORITHMS						Core
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
	L	T	D	P			
Data Structures	3	-	-	-	40	60	3

Course Objectives

Develop ability to

1. Acquire the knowledge of Algorithm and problem-solving technique, limitations of algorithms.
2. Understand different techniques like divide and conquer, transfer and conquer etc., to solve problems.
3. Understand different techniques like greedy method and dynamic programming
4. Understand the concepts of NP-Hard and NP-Complete.
5. Illustrating the methods of backtracking and branch bound techniques to solve the problems like N-queens problem, graph coloring.

Course Outcomes

At the end of the course, the student would be able to:

1. Identify asymptotic notations and basic efficiency classes.
2. Solve problems using various techniques like divide-and-conquer and transfer-and-conquer.
3. Use different algorithms like TSP, Floyd's to solve real world problems.
4. Analyze the P and NP classes of problems.
5. Develop solutions for n - Queens problem, Subset – Sum Problem, Assignment problem, Knapsack problem.

UNIT - I

Introduction: Algorithm, Fundamentals of algorithmic problem solving, Fundamentals of the analysis of algorithm efficiency, Asymptotic Notations and basic efficiency classes, Mathematical Analysis of Non-Recursive and Recursive Algorithms, The substitute method, Recursion tree method, Master method.

UNIT - II

Divide and Conquer: General Method, Binary Search, finding minimum and maximum Merge Sort analysis, Quick Sort analysis, Strassen's matrix multiplication.

Transfer and conquer: Introduction, Balanced search trees, Heap and Heap sort.

UNIT - III

The greedy method: The General Method, Knapsack problem, Job Sequencing with Deadlines, Minimum- Cost Spanning Trees: Prim's Algorithm, Kruskal's Algorithm, Single Source Shortest Paths.

Dynamic programming: The General Method, multistage graph, Warshall's Algorithm, Floyd's Algorithm for the All-Pairs Shortest Paths Problem, single source shortest path, The Travelling Salesperson problem, optimal binary search,0/1 knapsack.

UNIT – IV

Pattern Matching: The naïve string-matching algorithm, Brute Force String Matching, KMP algorithm.

NP-Hard and NP-Complete Problems: Basic concepts: non-deterministic algorithms, the classes NP, NP Complete and NP Hard problems, clique decision problem, chromatic number decision problem, Cook's theorem.

UNIT V

Backtracking: n - Queens Problem, Subset – Sum of Subsets Problem, graph coloring.

Branch and bound: Assignment problem, Knapsack problem, 15 puzzle problem, travelling salesman problem

Tries: Standard Tries, Compressed Tries, Suffix Tries, Search Engine Indexing.

Suggested Readings:

1. "Introduction to The Design and Analysis of Algorithms", Anany Levitin, Pearson Education, Delhi, 2nd Edition, 2007, ISBN: 9780321358288.
2. "Fundamentals of Computer Algorithms", Ellis Horowitz, Sartaj Sahni, Sanguthevar Rajasekaran: Universities Press, Hyderabad, 2nd Edition, 2007, ISBN: 10: 8173716129.
3. "Introduction to Algorithms", Thomas H. Cormen, Charles E. Leiserson, Ronal L. Rivest, Clifford Stein: PHI, London, England, 3rd Edition, 2010, ISBN: 9780262033848.
4. "Introduction to the Design and Analysis of Algorithms A Strategic Approach", R.C.T. Lee, S.S. Tseng, R.C. Chang and Y.T. Tsai: McGraw-Hill Higher Education, USA, International Edition, 2005, ISBN-13: 978-0071243469.

Course Code	Course Title						Core/Elective
U23CM502	ARTIFICIAL INTELLIGENCE						Core
Prerequisite	Contact Hours per Week				CIE	SEE	Credits
	L	T	D	P			
Data Structures	3	-	-	-	40	60	03

Course Objectives

The course will introduce the students to

1. Understand the various characteristics of Intelligent agents.
2. Learn the different search strategies in AI.
3. Learn to represent knowledge in solving AI problems.
4. Understand the different ways of designing software agents.
5. Know about the various applications of AI.

Course Outcomes

After successful completion of the course the students will be able to

1. Use appropriate search algorithms for any AI problem.
2. Represent a problem using first order and predicate logic.
3. Provide the apt agent strategy to solve a given problem.
4. Design software agents to solve a problem.
5. Design applications for NLP that use Artificial Intelligence.

UNIT – I

Introduction–Definition - Future of Artificial Intelligence –Characteristics of Intelligent Agents– Typical Intelligent Agents – Problem Solving Approach to Typical AI problems.

UNIT – II

Problem Solving Methods

Problem solving Methods - Search Strategies- Uninformed - Informed - Heuristics - Local Search Algorithms and Optimization Problems - Searching with Partial Observations - Constraint Satisfaction Problems – Constraint Propagation - Backtracking Search - Game Playing - Optimal Decisions in Games– Alpha - Beta Pruning - Stochastic Games

UNIT – III

Knowledge Representation

First Order Predicate Logic – Unification – Forward Chaining-Backward Chaining – Resolution – Knowledge Representation - Ontological Engineering-Categories and Objects – Events - Mental Events and Mental Objects - Reasoning Systems for Categories - Reasoning with Default Information.

UNIT – IV

Software Agents

Architecture for Intelligent Agents – Agent communication – Negotiation and Bargaining – Argumentation among Agents – Trust and Reputation in Multi-agent systems.

UNIT – V

Applications of AI

AI applications – Language Models – Information Retrieval- Information Extraction – Natural Language Processing - Machine Translation – Speech Recognition – Robot – Hardware – Perception – Planning – Moving

Suggested Readings:

1. S. Russell and P. Norvig, "Artificial Intelligence: A Modern Approach‖, Prentice Hall, Third Edition, 2009.
2. I. Bratko, Prolog Programming for Artificial Intelligence‖, Fourth edition, Addison-Wesley Educational Publishers Inc., 2011.
3. M. Tim Jones, Artificial Intelligence: A Systems Approach(Computer Science)‖, Jones and Bartlett Publishers, Inc.; First Edition, 2008
4. Nils J. Nilsson, ―The Quest for Artificial Intelligence‖, Cambridge University Press, 2009.
5. William F. Clocksin and Christopher S. Mellish,‖ Programming in Prolog: Using the ISO Standard‖, Fifth Edition, Springer, 2003.
6. Gerhard Weiss, ―Multi Agent Systems‖, Second Edition, MIT Press, 2013.
7. David L. Poole and Alan K. Mackworth, ―Artificial Intelligence: Foundations of Computational Agents‖, Cambridge University Press, 2010.

Course Code	Course Title						Core/Elective
U23IT502	FULL STACK DEVELOPMENT						CORE
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
Java Programming	L	T	D	P			
	3	-	-	-	40	60	3

Course Objectives

Develop ability to

1. Work with Webserver and Java Script
2. Gain knowledge of HTTP services and the DNS Modules.
3. Work with MongoDB and the associated connection objects.
4. Learn the Express and the Angular JS.
5. Understand the ReactJS and its Lifecycle.

Course Outcomes

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

1. Understand Full stack components for developing web application.
2. Apply packages of NodeJS to work with Data, Files, Http Requests and Responses.
3. Use MongoDB data base for storing and processing huge data and connects with NodeJS application.
4. Design faster and effective single page applications using Express and Angular.
5. Create interactive user interfaces with react components.

UNIT-I

Introduction to Full Stack Development:

Understanding the Basic Web Development Framework- User, Browser, Webserver, Backend Services, Full Stack Components - Node.js, MongoDB, Express, React, Angular. Java Script Fundamentals, NodeJS- Understanding Node.js, Installing Node.js, Working with Node Packages, creating a Node.js Application, Understanding the Node.js Event Model, Adding Work to the Event Queue, Implementing Callbacks.

UNIT-II

Node.js:

Working with JSON, Using the Buffer Module to Buffer Data, Using the Stream Module to Stream Data, Accessing the File System from Node.js- Opening, Closing, Writing, Reading Files and other File System Tasks. Implementing HTTP Services in Node.js- Processing URLs, Processing Query Strings and Form Parameters, Understanding Request, Response, and Server Objects, Implementing HTTP Clients and Servers in Node.js, Implementing HTTPS Servers and Clients. Using Additional Node.js Modules-Using the OS Module, Using the Util Module, Using the DNS Module, Using the crypto Module.

UNIT-III

MongoDB:

Need of NoSQL, Understanding MongoDB, MongoDB Data Types, Planning Your Data Model, Building the MongoDB Environment, Administering User Accounts, Configuring Access Control, Administering Databases, Managing Collections, Adding the MongoDB Driver to Node.js, Connecting to MongoDB from Node.js, Understanding the Objects Used in the MongoDB Node.js Driver, Accessing and Manipulating Databases, Accessing and Manipulating Collections

UNIT-IV

Express and Angular:

Getting Started with Express, Configuring Routes, Using Requests Objects, Using Response Objects. Angular: importance of Angular, Understanding Angular, creating a Basic Angular Application, Angular Components, Expressions, Data Binding, Built-in Directives, Custom Directives, Implementing Angular Services in Web Applications.

UNIT-V

React:

Need of React, Simple React Structure, The Virtual DOM, React Components, Introducing React Components, Creating Components in React, Data and Data Flow in React, Rendering and Life Cycle Methods in React, Working with forms in React, integrating third party libraries, Routing in React.

Suggested Reading:

1. Brad Dayley, Brendan Dayley, Caleb Dayley., Node.js, MongoDB and Angular Web Development, 2nd Edition, Addison-Wesley, 2019.
2. Mark Tielens Thomas, React in Action, 1st Edition, Manning Publications.

Course Code	Course Title						Core/Elective
U23IT5L1	FULL STACK DEVELOPMENT LAB						CORE
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
	L	T	D	P			
Java Programming	-	-	-	3	25	50	1.5

Course Objectives

Develop ability to

1. Build software development skills using java programming for real world applications.
2. Implement frontend and backend of an application
3. Create Java application programs using sound OOP practices such as interfaces, exception handling multithreading.
4. Understand fundamentals of object-oriented programming in Java.
5. Implement classical problems using java programming.

Course Outcomes

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

1. Design flexible and responsive Web applications using Node JS, React, Express and Angular.
2. Perform CRUD operations with MongoDB on huge amount of data.
3. Develop real time applications using react components.
4. Use various full stack modules to handle http requests and responses.
5. Implement the components of ReactJS.

List of Experiments

1. Create an application to setup Node JS environment and display “Hello World”.
2. Create a Node JS application for user login system.
3. Write a Node JS program to perform read, write and other operations on a file.
4. Write a Node JS program to read form data from query string and generate response using NodeJS.
5. Create a food delivery website where users can order food from a particular restaurant listed in the website for handling http requests and responses using NodeJS.
6. Implement a program with basic commands on databases and collections using MongoDB.
7. Implement CRUD operations on the given dataset using MongoDB.
8. Perform Count, Limit, Sort, and Skip operations on the given collections using MongoDB.
9. Develop an angular JS form to apply CSS and Events.
10. Write a program to create a voting application using React JS.

Suggested Readings:

1. Brad Dayley, Brendan Dayley, Caleb Dayley., Node.js, MongoDB and Angular Web Development, 2nd Edition, Addison-Wesley,2019.
2.Mark Tielens Thomas., React in Action, 1st Edition, Manning Publications.

Course Code	Course Title						Core/Elective
U24CM5L2	ARTIFICIAL INTELLIGENCE LAB						Core
Prerequisite	Contact Hours per Week				CIE	SEE	Credits
	L	T	D	P			
Python	-	-	-	3	25	50	1.5

Course Objectives:

The objective of this lab is to get an overview of the various Artificial Intelligence techniques and can able to demonstrate those using python.

1. Introduce students to the basic concepts of AI Searching techniques
2. Develop skills for solving practical problems.
3. Gain experience neural networks.
4. Perform experiments in Machine Learning using real-world data. ·
5. Implement classifiers and Regression algorithm.

Course Outcomes:

After the completion of the course the student can able to:

1. Design and implement AI solutions searching techniques using AI.
2. Know about facts of querying.
3. Apply tree mechanism using AI with Neural Network
4. Perform experiments in Machine Learning using real-world data. ·
5. Implement classifiers and Regression algorithm

List of Experiments:

1. Develop a Python program to perform Breadth-First Search (BFS).
2. Create a Python program that carries out Depth-First Search (DFS).
3. Build a console-based Tic-Tac-Toe game using Python.
4. Write a Python script to solve the 8-Puzzle problem.
5. Design a Python program to address the classic Water Jug puzzle.
6. Implement the Travelling Salesman Problem (TSP) using Python.
7. Construct a recursive Python solution for the Tower of Hanoi problem.
8. Simulate the Monkey and Banana problem through a Python implementation.
9. Write a Python solution for the Missionaries and Cannibals puzzle.
10. Implement the N-Queens problem using backtracking in Python.
11. Write a Python program to train and evaluate the following machine learning models using scikit-learn:
    (i) Decision Tree Classifier
    (ii) Multi-layer Perceptron (Feedforward Neural Network.
    (iii) Gaussian Naive Bayes classifier using scikit-learn (Implement any two classifiers).
12. Create a Python program that performs Linear Regression using any two different algorithms.

Suggested Readings:

1. S. Russell and P. Norvig, "Artificial Intelligence: A Modern Approach‖, Prentice Hall, Third Edition, 2009.
2. M. Tim Jones, ―Artificial Intelligence: A Systems Approach (Computer Science), Jones and Bartlett Publishers, Inc.; First Edition, 2008.
3. Elaine Rich, Kevin Knight, and Shivashankar B. Nair, Artificial Intelligence, McGraw-Hill Education, Third Edition, 2008.
4. Gerhard Weiss, ―Multi Agent Systems‖, Second Edition, MIT Press, 2013.
5. David L. Poole and Alan K. Mackworth, ―Artificial Intelligence: Foundations of Computational Agents‖, Cambridge University Press, 2010.

Course Code	Course Title						Core/Elective
U23IT5P1	INTERNSHIP						CORE
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
Programming Languages	L	T	D	P			
	-	-	-	2	50	-	1

Course Objectives

Develop ability to

1. Produce an accurate record of work performed during the Internship
2. Apply engineering knowledge to a problem in industry
3. Produce a technical report
4. Discuss work in a team environment, if relevant to the project
5. Conduct herself/himself responsibly, safely, and ethically in a professional environment

Course Outcomes

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

1. Design and develop a small and simple product in hardware or software.
2. Complete the task or realize a pre specified target, with a specified scope.
3. Learn to find alternate viable solutions for a given problem and evaluate these alternatives with reference to pre specified criteria.
4. Gain knowledge of working practices within industrial / R&D environments.
5. Implement the selected solution and document the same.

Guidelines:

Internship is introduced as part of the curriculum for encouraging students to work on problems of interest to industries. A batch of three students will be attached to a person from the Government or Private Organizations /Computer Industry/Software Companies/R&D Organization for a period of 4 weeks. This will be during the summer vacation following the completion of the IV Semester. One faculty coordinator will also be attached to the group of 3 students to monitor the progress and to interact with the industry coordinate (person from industry). The course schedule will depend on the specific internship/training experience. The typical time per topic will vary depending on the internship

1. Overview of company/project
2. Safety training
3. Discussions with project teams
4. Background research, review of documents, white papers, and scientific papers
5. Planning, designing, and reviewing the planned work
6. Executing the plans
7. Documenting progress, experiments, and other technical documentation
8. Further team discussions to discuss results
9. Final report writing and presentation

After the completion of the internship, each student will be required to:

1. Submit a brief technical report on the internship undergone.
2. Present the work through a seminar talk (to be organized by the Department)

Internship will be evaluated for 50 marks as CIE based on Punctuality in Attendance (10), Maintenance of Diary on daily work done (20), seminar presentation / viva voce followed by a report submission (20) to a committee consisting of Head of the Department, Senior Faculty and Faculty Advisor / Supervisor for evaluation.

Note: Students have to undergo summer internship of 4 weeks at the end of semester IV and credits will be awarded after evaluation in V semester.

LORDS INSTITUTE OF ENGINEERING AND TECHNOLOGY
(An Autonomous Institution)
DEPARTMENT OF INFORMATION TECHNOLOGY
SCHEME OF INSTRUCTIONS & EXAMINATIONS [LR-23]
(W.E.F Academic Year 2025-26)
B.E. VI-Semester

S. No.	Course Code	Course Category	Course Title	Scheme of Instructions				Scheme of Examination			CREDITS
				L	T	P/ D	Contact Hours/Week	CIE	SEE	Duration in Hours	
Theory Course											
1	U23IT601	PCC	Embedded Systems and Internet of Things	3	-	-	3	40	60	3	3
2	U23CD602	PCC	Computer Networks	3	-	-	3	40	60	3	3
3	U23CM602	PCC	Machine Learning	3	-	-	3	40	60	3	3
	--	PEC	Professional Elective – I	3	-	-	3	40	60	3	3
	--	OEC	Open Elective – II	3	-	-	3	40	60	3	3
Practical / Laboratory Course											
7	U23IT6L1	PCC	Embedded Systems and Internet of Things Lab	-	-	3	3	25	50	3	1.5
8	U23CD6L2	PCC	Computer Networks Lab	-	-	3	3	25	50	3	1.5
9	U23CM6L2	PCC	Machine Learning Lab	-	-	3	3	25	50	3	1.5
Project											
10	U23IT6P1	PROJ	Mini Project	-	-	6	6	50	50	3	2
Total				**15**	**-**	**15**	**30**	**325**	**500**	**-**	**21.5**

L: Lecture*(Hrs/Wk/Sem)* **T**: Tutorial *(Hrs/Wk/Sem)* **P**: Practical / **D**: Drawing *(Hrs/Wk/Sem)*
CIE: Continuous Internal Evaluation **SEE**: Semester End Examination (Univ.Exam)
OEC: Open Elective Courses **IT:** Information Technology
PCC: Programme Core Courses **PEC**: Professional Elective
PROJ: Project

Note:
1. Each contact hour is a Clock Hour.
2. The duration of the practical class is three hours, however it can be extended wherever necessary, to enable the student to complete the experiment.

Course Code	Course Title						Core/Elective
U23IT601	EMBEDDED SYSTEMS & INTERNET OF THINGS						CORE
Prerequisite	Hours Per Week				CIE	SEE	Credits
COM	L	T	D	P			
	3	-	-	-	40	60	3

Course Objectives

Develop ability to

1.Learn the internal architecture and programming of an embedded processor.
2. Introduce interfacing I/O devices to the processor.
3. Introduce the evolution of the Internet of Things (IoT).
4. Build a small low-cost embedded and IoT system using Arduino/Raspberry Pi/open platform.
5.Apply the concept of Internet of Things in real world scenario

Course Outcomes

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

1. Learn the internal architecture and programming of an embedded processor.
2. Interface I/O devices to the processor.
3. Present the Internet of Things' (IoT) progress.
4. Use the Arduino, Raspberry Pi, and open platform, to construct a small, inexpensive embedded and Internet of Things system.
5. Use the Internet of Things concept in a practical setting.

UNIT – I

8-Bit Embedded Processor: 8-Bit Microcontroller, Architecture, Instruction Set and Programming, Programming Parallel Ports, Timers and Serial Port, Interrupt Handling.

UNIT – II

Embedded C Programming: Memory And I/O Devices Interfacing, Programming Embedded Systems in C, Need For RTOS, Multiple Tasks and Processes, Context Switching, Priority Based Scheduling Policies.

UNIT- III

IoT and Arduino Programming: Introduction to the Concept of IoT Devices, IoT Devices Versus Computers, IoT Configurations, Basic Components, Introduction to Arduino, Types of Arduino, Arduino Toolchain, Arduino Programming Structure, Sketches, Pins Input/Output from Pins Using Sketches, Introduction to Arduino Shields, Integration of Sensors and Actuators with Arduino.

UNIT – IV

IoT Communication and Open Platforms: IoT Communication Models and APIs, IoT Communication Protocols, Bluetooth, WiFi, ZigBee, GPS, GSM modules, Open Platform (like Raspberry Pi), Architecture, Programming, Interfacing, Accessing GPIO Pins, Sending and Receiving Signals Using GPIO Pins, Connecting to the Cloud.

UNIT - V

Applications Development: Complete Design of Embedded Systems, Development of IoT Applications, Home Automation, Smart Agriculture, Smart Cities, Smart Healthcare.

Suggested Readings:

1. Raj Kamal, Embedded Systems – Architecture, Programming and Design, 2nd Edition, TMH, 2008.
2. Andrew N. Sloss, Dominic Symes, Chris Wright, ARM Systems Developer's Guides – Designing & Optimizing System Software, Elsevier, 2008.
3. Vijay Madisetti and Arshdeep Bahga, Internet of Things: A Hands-On Approach, VPT edition1, 2014.

Course Code	Course Title						Core / Elective
U23CD504, U23CD602	COMPUTER NETWORKS						Professional Elective & Core
Prerequisite	Contact Hours per Week				CIE	SEE	Credits
	L	T	D	P			
C	3	-	-	-	40	60	3

Course Objectives

1. To introduce the fundamental various types of computer networks.
2. To demonstrate the TCP/IP and OSI models with merits and demerits.
3. To explore the IP Addressing Mechanisms
4. To Understand the World Wide Web concepts.
5. Classify the routine protocols and analyze how to assign the IP address for the given network.

Course Outcomes:

1. Explain & design the various reference models and networks.
2. Identify the different types of network devices and Multiple Access Protocols.
3. Use IP addressing Scheme and to interconnect various networks and Routing mechanism
4. Explain transport layer protocols: TCP, UDP.
5. Explain and use various application layer protocols: HTTP, DNS, and SMTP, FTP etc

UNIT-I

Introduction: Network Uses, Topologies, Transmission Modes, Network Hardware, Network Software, Reference Models:OSI, TCP/IP.

The Physical Layer: Theoretical basis for communication, Guided transmission media, Wireless transmission.

UNIT-II

The Data Link Layer: Design issues, error detection and correction, elementary data link protocols, sliding windowprotocols, example data link protocols – HDLC.

The Medium Access Control Sublayer: The Channel allocations problem, multiple access protocols, Ethernet, WirelessLANs.

UNIT-III

The Network Layer: Network layer design issues, routing algorithms, Congestion control algorithms.

Internetworking: Concatenated virtual circuits, Connectionless internet working, Tunneling, the network layer in theinternet: IP protocol, IP addresses, OSPF, BGP, (IPv4 and IPv6).

UNIT-IV

Network Programming: Socket Interface: Sockets, Socket Address, Elementary Sockets, Advanced Sockets.

The Transport Layer: Transport service, Elements of transport protocols, Simple Transport Protocol, Internet transportlayer protocols: UDP and TCP.

UNIT-V

The Application Layer: Domain Name System-, Electronic Mail-Architecture and Services, World Wide Web: architecturaloverview, dynamic web document and http.

Text Books:

1. Andrew S. Tanenbaurn, Computer Networks, Fourth Edition, Pearson Education.
2. W. Richard stevens, Unix Network Programming‖ Prentice Hall/PearsonEducation, 2009.
3. James F. Kurose, Keith W, Ross, Computer Networking, Atop-Down Approach Featuring the Internet, Third Edition,Pearson Education, 2005.

Course Code	Course Title						Core/Elective
U23CM501 U23CM602	MACHINE LEARNING						Core
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
	L	T	D	P			
Artificial Intelligence	3	-	-	-	40	60	3

Course Objectives:

This course will enable students to

1. To introduce students to the basic concepts of Data Science and techniques of Machine Learning.
2. To develop skills of using recent machine learning software for solving practical problems.
3. To gain experience of doing independent study and research.
4. To develop an understanding of the role of machine learning in massive scale automation.
5. To design and implement various machine learning algorithms in the range of real world applications

Course Outcomes:

On completion of this course, the students are able to :

1. Design and implement machine learning solutions of classification, regression problems.
2. Evaluate and interpret the results of the machine learning algorithms.
3. Evaluate exploratory data analysis and Data preparation and preprocessing on different datasets.
4. Calculate Statistical measurements of the given data.
5. Analyze and identify the best algorithm matches for a given dataset.

UNIT – I

Introduction: What is Machine Learning, Use Machine Learning, and Types of Machine Learning Systems: supervised, unsupervised, semi-supervised, Reinforcement Learning, Batch and Online Learning, Main Challenges of Machine Learning.

UNIT – II

Descriptive Statistics: Data representation, types of data- nominal, ordinal, interval and continuous, central tendency- calculating mean mode median, mean vs median, variability, variance, standard deviation, Mean Absolute Deviation using sample dataset, finding the percentile, interquartile range, Box Plot, Outlier, whisker, calculating correlation, covariance, causation. Exploratory data analysis, Data preparation and preprocessing, Data visualization.

UNIT – III

Regression: Introduction to Regression analysis, measure of linear relationship, Regression with stats models, Determining coefficient, meaning and significance of coefficients, coefficient calculation with least square method, Types of regression, Simple Linear Regression, Using Multiple features, Polynomial Regression, Metrics for Regression: MSE, RMSE, MAE.

UNIT – IV

Classification: Classification problem, Probability based approach, Logistic Regression- log-odd, sigmoid transformation, Metrics: Confusion Matrix, Accuracy, Error Rate, Precision, Recall, ROC curve, F1 score, and introduction to gradient descent.

UNIT – V

Non Parametric & SVM classification: About Non parametric classification, Decision Trees: Entropy, Gain ratio, Information Gain, Splitting criteria.
Ensemble Method: Introduction to Random Forest, Accuracy measure & performance
Instance based learning- Introduction, KNN algorithm, Distance measures, model building, locally weighted regression, radial basis functions, SVM classifier, hyper-plane, slack variables, geometric transformation kernel trick, kernel transformation.

Suggested Readings:

1. Machine Learning in Action, Peter Harrington, Dreamtech Press India Pvt. Ltd, 1st Edition,2012.
2. The Field Guide to Data Science ,Booz, Allen, Hamilton, Manning Publications 2nd Edition,2018.
3. Hands-On Machine Learning with Scikit-Learn and TensorFlow Aurelian Géron, ,O'Reilly Media, 3rd Edition,2017.
4. Pattern Recognition and Machine Learning, A foundational book providing a deep theoretical understanding of machine learning models using statistical techniques, Christopher M. Bishop, Springer, 1st Edition, 2006.
5. Python Machine Learning, Sebastian Raschka and Vahid Mirjalili, A practical guide focusing on machine learning techniques using Python, Scikit-learn, Keras, and TensorFlow., Packt Publishing, 3rd Edition, 2019

Course Code	Course Title						Core/Elective
U23IT6L1	EMBEDDED SYSTEMS & INTERNET OF THINGS LAB						CORE
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
DLD	L	T	D	P			
	-	-	-	3	25	50	1.5

Course Objectives

Develop ability to

1.Familiarize with interfacing LEDs and switches using 8051 Microcontroller.
2.Aquaint with interface controls using 8051.
3.Explore design and development of an embedded system.
4.Know the interfacing programs using Python.
5.Understand the applications using Raspberry Pi and Arduino.

Course Outcomes

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

1. Possess the passion for acquiring programming skills in using different tools.
2.Able to design and develop embedded systems (hardware, peripherals and firmware).
3.Write code for different forms of interfacing devices.
4.Develop python programs that run on Raspberry Pi4.
5.Interface Sensors and Actuators with Raspberry Pi4.

List of Experiments

A. Interfacing Programs using embedded C on ARM Microcontroller Kit:

1. Program to perform arithmetic operation (addition ,subtract ,multiply, division).
2. Program to interface8-Bit LED and switch interface.
3. Program to implement Buzzer interface on IDE environment.
4. Stepper motor interface.
5. ADC &Temperature sensor LM35interface.
6. Program to produce show the Time delay.

B. Internet of Things (IoT) Experiments to write and test on Raspberry Pi.

1.Switching LED on/off from Raspberry Pi Console.
2.Interfacing an LED and Switch with Raspberry Pi.
3.Interfacing a Light Sensor with Raspberry Pi.
4.Interfacing Smoke Sensor to give alert message to fire department.
5.Implementation of Traffic Light System based on density, to decrease congestion.
6.Implementation of Home Automation System using WiFi Module.

Suggested Readings:

1.Kenneth J.Ayala, “The 8051 Microcontroller”, 3rd Edition, Thomson, 2014.
2.ArshdeepBahga, Vijay Madisetti, “Internet of Things: A Hands-on Approach”, Universities Press, 2014.

Course Code	Course Title						Core / Elective
U23CD6L2	COMPUTER NETWORKS LAB						Core
Prerequisite	Contact Hours per Week				CIE	SEE	Credits
	L	T	D	P			
C	-	-	-	3	25	50	1.5

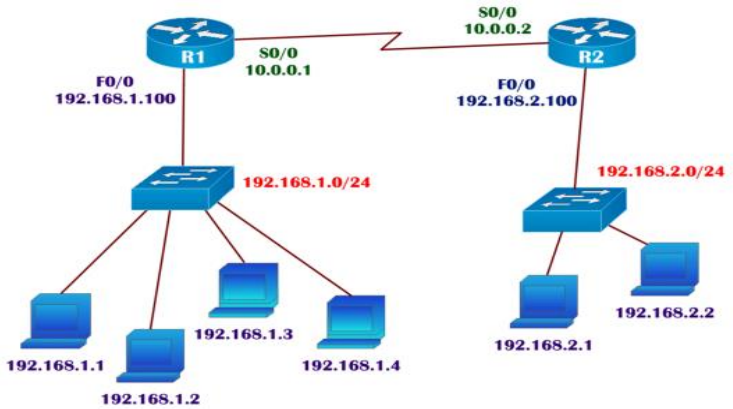
Course Objectives:

1. To familiarize students with the basic principles, protocols, and layered architecture of computer networks.
2. To reinforce the theory learned in the classroom via hands-on practice.
3. Develop practical skills in network configuration and management
4. To design and implement client-server applications using protocols such as TCP and UDP.
5. To develop simple socket programming applications in languages like C, Java, or Python.

Course Outcomes:

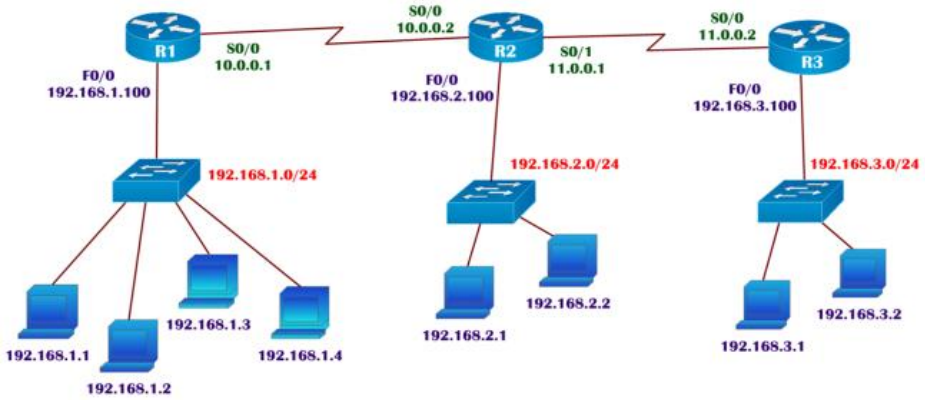
1. Explain and apply basic networking concepts, layered architecture, and protocol operations in practical scenarios.
2. Configure and troubleshoot network devices and design small-scale network topologies.
3. Develop and implement client–server communication using socket programming in TCP and UDP.
4. Simulate and analyze network protocols and topologies using simulation tools .
5. Capture, inspect, and analyze network traffic using tools such as Wireshark to understand protocol behavior and performance metrics.

Sr No	Experiment No.	NAME OF THE EXPERIMENT
01	Exp-01	**Case study about** a) Wireless LAN Configuration b) Cables – Coaxial, Twisted, Fiber Optic c) Cisco Router, Ports of router d) Cisco Switches & Types
02	Exp-02	**Build a Local Area Networks using Packet Tracer** a) Connect 4 computers in the LAN using Switch b) Configure IP addressing on all PC using 192.168.1.0/24 network c) Check connectivity between all the PÇ's using Ping command,traceroute
03	Exp-03	**Assigning & verifying IP Address to Router using Packet Tracer** a) Design the topology as per given diagram b) Configure IP Address as diagram & rules Verify the interface status using commands



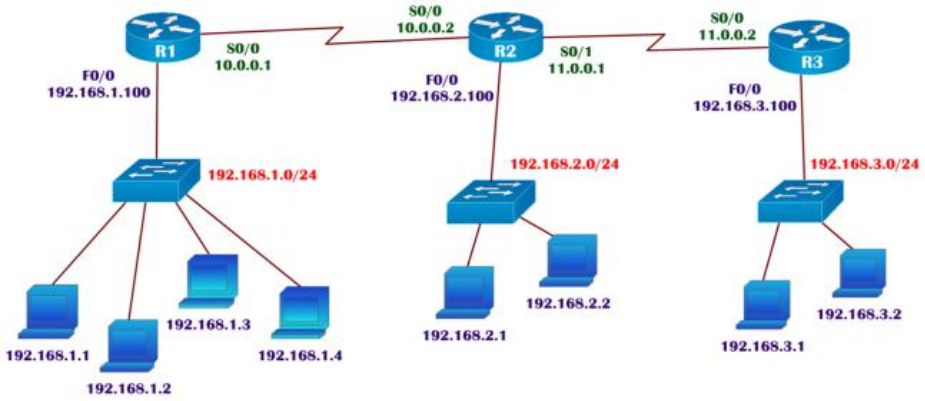
04 Exp-04 Configuring Static Routing using Packet Tracer

a) Design the topology
b) Assign IP address according to diagram
c) Make sure that interfaces used should be in UP UP state
d) Configure static routing
e) Verify routing table and reachability between LAN's (Using PING & TRACE commands)



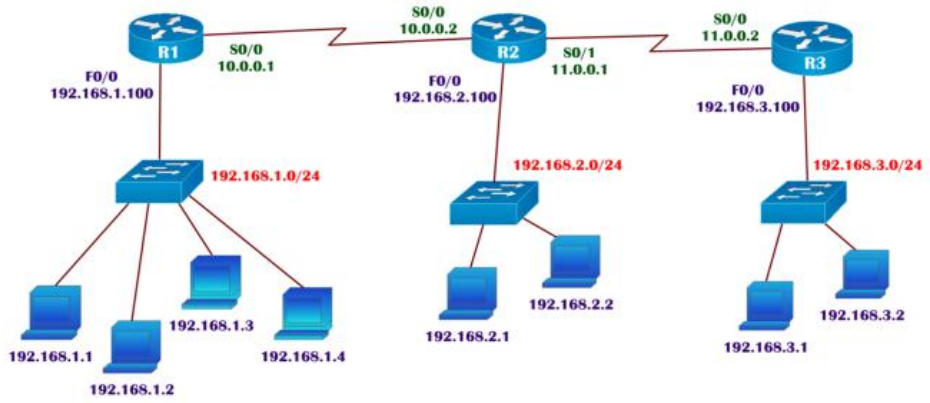
05 Exp-05 Configuring Default Routing using Packet Tracer

a) Design the topology
b) Assign IP address according to diagram
c) Make sure that interfaces used should be in UP-UP state
d) Configure Default route used on R1 & R3, Static routing on R2
e) Verify routing table and reachability between LAN's (Using PING & TRACE commands)



06 **Exp-06** **Configuring Dynamic Routing using Packet Tracer**

a) Design the topology
b) Assign IP address according to diagram
c) Make sure that interfaces used should be in UP UP state
d) Configure Dynamic Routing using EIGRP 100
e) Verify routing table and reachability between LAN's (Using PING & TRACE commands)



07 **Exp-07** **Virtual LAN Configuration on Switches**

a) Case study – Creating VLAN, Assigning Ports
b) Create four VLAN's (VLAN 10, 20, 30, 40)
c) Configure port fa0/8 in to VLAN 10
d) Configure multiple ports (4-7 and 10) into vlan 20

08 **Exp-08**

a) Case study - Proxy server, Web server
b) Connect all PC's and Printer in LAN in real scenario and test reachability using PING Command

Course Code	Course Title						Core/Elective
U23CM5L1, U23CM6L2	MACHINE LEARNING LAB						Core
Prerequisite	Contact Hours per Week				CIE	SEE	Credits
	L	T	D	P			
Python Programming	-	-	3	3	25	50	1.5

Course Objectives:

This course enable students to:

1. To introduce students to the basic concepts of Data Science and techniques of Machine Learning.
2. To develop skills of using recent machine learning software for solving practical problems.
3. To gain experience of doing independent study and research.
4. Be capable of confidently applying common Machine Learning algorithms in practice and implementing their own
5. Be capable of performing experiments in Machine Learning using real-world data.

Course Outcomes:

On completion of this course, the students are able to:

1. The student must be able to design and implement machine learning solutions to classification, regression problems.
2. Understand complexity of Machine Learning algorithms and their limitations
3. Able to evaluate and interpret the results of the algorithms.
4. Implement Decision trees and various algorithms
5. Implement and Analyze various random forest techniques.

1. Implement a program to demonstrate the following
   a) Operation of data types in Python.
   b) Different Arithmetic Operations on numbers in Python.
   c) Create, concatenate and print a string and access substring from a given string.
   d) Append, and remove lists in python.
   e) Demonstrate working with tuples in python.
   f) Demonstrate working with dictionaries in python.

2. Using python write a NumPy program to compute the
   a) Expected Value
   b) Mean
   c) Standard deviation
   d) Variance
   e) Covariance
   f) Covariance Matrix of two given arrays.

3. For a given set of training data examples stored in a .CSV file, demonstrate Data Preprocessing in Machine learning with the following steps
   a) Getting the dataset.
   b) Importing libraries.
   c) Importing datasets.
   d) Finding Missing Data.
   e) Encoding Categorical Data.
   f) Splitting dataset into training and test set.
   g) Feature scaling.

4. Build a linear regression model using python for a particular data set by
   a) Splitting Training data and Test data.
   b) Evaluate the model (intercept and slope).
   c) Visualize the training set and testing set
   d) predicting the test set result
   e) compare actual output values with predicted values

5. The dataset contains information of users from a company's database. It contains information about UserID, Gender, Age, EstimatedSalary, and Purchased. Use this dataset for predicting that a user will purchase the company's newly launched product or not by Logistic Regression model.

User ID	Gender	Age	EstimatedSalary	Purchased
15624510	Male	19	19000	0
15810944	Male	35	20000	0
15668575	Female	26	43000	0
15603246	Female	27	57000	0
15804002	Male	19	76000	0
15728773	Male	27	58000	0
15598044	Female	27	84000	0
15694829	Female	32	150000	1
15600575	Male	25	33000	0
15727311	Female	35	65000	0
15570769	Female	26	80000	0
15606274	Female	26	52000	0
15746139	Male	20	86000	0
15704987	Male	32	18000	0
15628972	Male	18	82000	0
15697686	Male	29	80000	0
15733883	Male	47	25000	1
15617482	Male	45	26000	1
15704583	Male	46	28000	1

6. Implement a program to demonstrate the working of the decision tree based ID3 algorithm. Use an appropriate data set for building the decision tree and apply this knowledge to classify a new sample.

7. Implement k-nearest neighbor's classification to classify the iris data set using python.

8. Given the following data, which specify classifications for nine combinations of VAR1 and VAR2 predict a classification for a case where VAR1=0.906 and VAR2=0.606, using the result of k-means clustering with 3 means (i.e., 3 centroids)

VAR1	VAR2	CLASS
1.713	1.586	0
0.180	1.786	1
0.353	1.240	1
0.940	1.566	0
1.486	0.759	1
1.266	1.106	0
1.540	0.419	1
0.459	1.799	1
0.773	0.186	1

9. Evaluate the metrics for all types of machine learning algorithms using sample data.

10. Implement an algorithm to demonstrate the significance of SVM.

Suggested Readings:

1. The Field Guide to Data Science ,Booz, Allen, Hamilton,Manning Publications 2nd Edition,2018.
2. Hands-On Machine Learning with Scikit-Learn and TensorFlow,Aurélien Géron, O'Reilly Media, 1st Edition,2017.
3. Machine Learning in Action, Peter Harrington, Manning Publications.2012.
4. Pattern Recognition and Machine Learning, A foundational book providing a deep theoretical understanding of machine learning models using statistical techniques, Christopher M. Bishop, Springer, 1st Edition, 2006.
5. Python Machine Learning, Sebastian Raschka and Vahid Mirjalili, A practical guide focusing on machine learning techniques using Python, Scikit-learn, Keras, and TensorFlow., Packt Publishing, 3rd Edition, 2019

Course Code	Course Title						Core/Elective
U23IT6P1	MINI PROJECT						CORE
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
Programming Languages	L	T	D	P			
	-	-	-	6	50	50	2

Course Objectives

Develop ability to

1. Enhance practical and professional skills.
2. Familiarize tools and techniques of systematic literature survey and documentation
3. Expose the students to industry practices and team work.
4. Encourage students to work with innovative and entrepreneurial ideas.
5. Make students evaluate different solutions based on economic and technical feasibility

Course Outcomes

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

1. Formulate a specific problem and give solution
2. Develop model/models either theoretical/practical/numerical form
3. Solve, interpret/correlate the results and discussions
4. Conclude the results obtained
5. Write the documentation in standard format

Guidelines:

1. As part of the curriculum in the VI- semester of the programme each student shall do a mini project, generally comprising about three to four weeks of prior reading, twelve weeks of active research, and finally a presentation of their work for assessment.
2. Four students will be allotted to one faculty supervisor for mentoring.
3. Mini projects should present students with an accessible challenge on which to demonstrate competence in research techniques, plus the opportunity to contribute something more original.
4. Mini projects shall have inter-disciplinary/ industry relevance.
5. The students can select a mathematical modelling based/Experimental investigations or Numerical modelling
6. All the investigations should be clearly stated and documented with the reasons/explanations.
7. The mini-project shall contain a clear statement of the research objectives, background of work, literature review, techniques used, prospective deliverables, and detailed discussion on results, conclusions and reference.
8. The CIE shall include reviews and the preparation of report consisting of a detailed problem statement and a literature review.
9. The preliminary results (if available) of the problem may also be discussed in the report.
10. The work has to be presented in front of the PRC committee which consists of one Supervisor and a minimum of two faculty members from the respective Department of the Institute.

Course Code	Course Title						Core/Elective
U23IT603	IMAGE PROCESSING						PEC-I
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
Programming for Problem Solving	L	T	D	P			
	3	-	-	-	40	60	3

Course Objectives

Develop ability to

1. Gain the fundamentals of digital image processing.
2. Provide mathematical foundations for digital manipulation of images, image acquisition, preprocessing, segmentation, Fourier domain processing and compression.
3. Formulate solutions to general image processing problems.
4. Understand and implement various image enhancement and restoration techniques in both spatial and frequency domains.
5. Develop the ability to extract meaningful features from images for analysis, representation, and recognition using modern image processing methods.

Course Outcomes

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

1. Understand the fundamental concepts of a digital image processing.
2. Evaluate the techniques for image enhancement and image restoration.
3. Categorize various compression techniques.
4. Interpret Image compression standards.
5. Interpret image segmentation and representation techniques.

UNIT-I

Fundamentals Digital image, Elements of digital geometry, Components of DIP, Visual detail. Visual preliminaries- Brightness adaptation and Contrast, Acuity and contour, Texture and pattern discrimination, Shape detection and recognition, Perception of color. Image formation- Geometric Model and Photometric Model.

UNIT-II

Image Enhancement Spatial Domain Methods –Binary Image, Negative of an Image, Log Transformations, Power law Transformation, contrast enhancement, Histogram equalization, Spatial Domain Filters- Smoothing filters, Sharpening filters. Frequency Domain Methods- Steps for filtering in the frequency domain, Smoothing filters, Sharpening filters.

UNIT-III

Image Restoration A model of the image degradation, noise models, restoration in the Presence of noise- spatial filtering, periodic noise reduction by frequency domain filtering, linear & position-invariant degradations, estimating the degradation function**.**

UNIT-IV

Segmentation Points detection, line detection, edge detection methods, Histogram based image segmentation, segmentation using split and merge method, region growing method, watershed method, k-means clustering method, self-similar fractal method.

UNIT-V

Representation, Description and Recognition Representation, boundary descriptors, regional descriptors, principal component analysis, relational descriptors. Recognition based on decision-theoretic and structural methods.

Suggested Reading:

1. Rafael C. Gonzalez and Richard E. Woods, *Digital Image Processing*, 4th Edition, Pearson.
2. B. Chanda and D. Dutta Majumder, *Digital Image Processing and Analysis*, 2nd Edition, PHI Learning.
3. Milan Sonka, Vaclav Hlavac, and Roger Boyle, *Image Processing, Analysis, and Machine Vision*, Cengage Learning.

Course Code	Course Title						Core/Elective
U23IT604	DATA VISUALIZATION						PEC-I
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
Python Programming	L	T	D	P			
	3	-	-	-	40	60	3

Course Objectives

Develop ability to

1. Learn different statistical methods for Data visualization.
2. Work with basics of R and Python.
3. Use Watson studio.
4. Visualize data effectively using R through various graphical techniques.
5. Learn functionalities and usages of Seaborn.

Course Outcomes

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

1. Apply statistical methods for Data visualization.
2. Gain knowledge on R and Python
3. Understand usage of various packages in R and Python.
4. Demonstrate knowledge of Watson studio.
5. Apply data visualization tools on various data sets.

UNIT I

Introduction to Statistics: Introduction to Statistics, Difference between inferential statistics and descriptive statistics, Inferential Statistics- Drawing Inferences from Data, Random Variables, Normal Probability Distribution, Sampling, Sample Statistics and Sampling Distributions. R overview and Installation- Overview and About R, R and R Studio Installation, Descriptive Data analysis using R, Description of basic functions used to describe data in R.

UNIT II

Data manipulation with R: Data manipulation packages, Data visualization with R. Data visualization in Watson Studio: Adding data to data refinery, Visualization of Data on Watson Studio.

UNIT III

Graphical Data Visualization in R: R Graphics Systems**,** Visualization of One Variable**,** Visualization of Two Variables, Multivariate and Advanced Plots, Customizing and Exporting Plots

UNIT IV

Data Visualization Tools in Python: Introduction to Matplotlib, Basic plots using matplotlib, Specialized Visualization Tools using Matplotlib, Advanced Visualization Tools using Matplotlib Waffle Charts, Word Clouds.

UNIT V

Introduction to Seaborn: Seaborn functionalities and usage, Spatial Visualizations and Analysis in Python with Folium, Case Study.

Suggested Readings:

1.Core Python Programming - Second Edition, R. Nageswara Rao, Dream tech Press.
2.R Graphics Essentials for Great Data Visualization by Alboukadel Kassambara.

Course Code	Course Title				Core/Elective		
U23IT605	MOBILE APPLICATION DEVELOPMENT				PEC-I		
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
	L	T	D	P			
Python Programming	3	-	-	-	40	60	3

Course Objectives

Develop ability to

1. Impart knowledge on Android OS design and Features.
2. Know the Android application components, user interface.
3. Implement data persistence techniques such as shared preferences, file storage, and content providers for mobile apps.
4. Explore mobile networking and telephony services, including background services, multithreading, and system broadcasts.
5. Compare Android and iOS development platforms, understanding the navigation, design, and data handling approaches in iOS.

Course Outcomes

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

1. Describe the Android platform, its features, architecture, and tools like ADB and ART.
2. Design basic building blocks of Android programming required for App development.
3. Apply different data persistence schemes for data storage mechanism in Android.
4. Develop background services and handle asynchronous tasks using multithreading and broadcast receivers.
5. Create and deploy location-based services using Maps API and geolocation features.

UNIT-I

Introduction to Android Operating System: Android OS and Features – Android development framework, Installing and running applications on Android Studio, Creating AVDs, Types of Android Application, Creating Activities, Activity Life Cycle, Activity states, monitoring state changes.

UNIT- II

Android application components: Android Manifest file, Externalizing recourses like Simple Values, Drawables, Layouts, Menus.

Building User Interfaces: Fundamental Android UI design, Layouts – Linear, Relative, Grid and Table Layouts. User Interface (UI) Components.

UNIT-III

Fragments: Creating fragments, Lifecycle of fragments, Fragment states, Adding fragments to Activity, adding, removing and replacing fragments with fragment transactions, interfacing between fragments and Activities.

UNIT-IV

Intents and Broadcasts: Using intents to launch Activities, Types of Intents, Passing data to Intents, Getting results from Activities, Broadcast Receivers – Using Intent filters to service implicit Intents, Resolving Intent filters.

UNIT-V

Database: Introduction to SQLite database, creating and opening a database, creating tables, inserting retrieving and deleting data. Creating content providers.

Suggested Readings:

1. Professional Android 4 Application Development, Reto Meier, Wiley India, (Wrox), 2012
2. Android Application Development for Java Programmers, James C Sheusi, Cengage Learning, 2013
3. Android Application Development (with Kitkat Support), Black Book, Pradeep Kothari, 2014, Dreamtech Press publisher, Kogent Learning Inc.,2014.

Course Code	Course Title						Core/Elective
U23IT606	SOFTWARE ENGINEERING						PEC-I
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
	L	T	D	P			
Programming for Problem Solving	3	-	-	-	40	60	3

Course Objectives

Develop ability to

1. Understand basic concepts of software development processes of defining a process framework.
2. Impart knowledge on various phases, methodologies and practices of software development.
3. Understand the concepts of quality standards and software change management.
4. Understand the importance of testing in software development and study various testing strategies.
5. Identify the risks and to study different estimation techniques.

Course Outcomes

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

1. Define different software development processes and their usability in different problem domains.
2. Explain the process of requirements collection, analyzing, and modeling requirements for effective understanding and communication with stake holders.
3. Design and Develop the architecture of real-world problems towards developing a blue print for implementation.
4. Understand the concepts of software equality, testing and maintenance.
5. Discuss the concepts related to Risk management and Software project Estimation

UNIT-I

Introduction to Software Engineering: A generic view of process, Software Engineering process framework, Nature of Software, Software Myths, Process Models.

Agile Development: Introduction to Agility and Agile Process, Agile Process Models.

UNIT-II

Understanding Requirements: Requirements Engineering, Establishing the Groundwork, Eliciting Requirements, Building the Requirement Model, Negotiating Requirements, Validating Requirements.

Design Concepts: Design within the Context of Software Engineering, the Design Process, Design Concepts, Design Principles, Concept of Structural Design, DFD's, Example Data flow diagrams.

Architectural Design: Software Architecture, Architecture Genres, Architecture Styles, Architecture Designs.

UNIT-III

Software Quality Assurance: Background Issues, Elements of Software Quality Assurance, SQA Tasks, Goals and Metrics, Formal Approaches to SQA, Statistical Software Quality Assurance, Software Reliability, The ISO 9000 Quality Standards, The SQA Plan.

Risk Management: Risk Strategies, Software Risks, Risk Identification, Risk Projection, Risk Refinement, Risk Mitigation, Monitoring, and Management, The RMMM Plan.

UNIT-IV

Testing Conventional Applications: Software Testing Fundamentals, Internal and External Views of Testing, White-Box Testing, Basis Path Testing, Control Structure Testing, Black–Box Testing.

Software Testing Strategies: A Strategic Approach to Software Testing, Strategic Issues, Test Strategies for Conventional Software, Validation Testing, System Testing, The Art of Debugging.

UNIT-V

Software Configuration Management: Software Configuration Management, The SCM Repository.
Product Metrics: A Frame work for Product Metrics, Metrics for the Requirements Model, Metrics for the Design Model, Metrics for Testing, Metrics for Maintenance.
Estimation: Software Project Estimation, Decomposition Techniques, Empirical Estimation Models, Specialized Estimation Techniques.
Software Process Improvement: The SPI Process, The CMMI.

Suggested Readings:

1. Software Engineering: A Practitioners Approach, Roger S.Pressman, McGrawHill, Eighth Edition, 2019.
2. Software Engineering, Ian Somerville, Ian Somerville, Pearson Education, Eighth Edition, 2007.
3. An Integrated Approach to Software Engineering, Pankaj Jalote, Narosa Publishing House, Third Edition 2008.

Course Code	Course Title						Core/Elective
U23IT503	DATA STRUCTURES						Open Elective-I
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
	L	T	D	P			
Programming for Problem Solving	3	-	-	-	40	60	3

Course Objectives

Develop ability to

1. Develop skills to design and analyze simple linear and non linear data structures, such as stacks, queues and Lists and their applications.
2. Gain programming skills to implement sorting and searching algorithms.
3. Strengthen the ability to identify and apply the suitable data structures for the given real world problem.
4. Gain knowledge in practical applications of data structures
5. Understand essential for future programming and software engineering courses.

Course Outcomes

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

1. Implement various data structures using arrays, linked lists.
2. Develop ADT necessary for solving problems based on Stacks and Queues.
3. Implement binary trees, general tree structures, advanced search trees, heaps, graphs.
4. Implement hash functions and handle collisions.
5. Implement various kinds of sorting techniques and apply appropriate techniques for solving a given problem.

UNIT – I

Introduction Data Structures and Algorithms: Introduction to data structures, classification of data structures, operations on data structures; Algorithm Specification, Recursive algorithms, Data Abstraction.
Performance analysis- Time Complexity and Space Complexity, Asymptotic Notation-Big O, Omega, and Theta notations.

UNIT – II

Stacks: Stack ADT, definition and operations, Implementations of stacks using array, applications of stacks, Arithmetic expression conversion and evaluation.
Queues: Queue ADT, definition and operations, Implementation of queues using Arrays, applications of linear queue, circular queue.

UNIT – III

Linked Lists: Introduction, Singly linked list, representation of a linked list in memory, operations on a Singly Linked List, Implementation of Singly Linked List.
Doubly Linked Lists: Operations on Doubly Linked List, Implementation of Doubly Linked List , Circular linked list, Implementation of Stack and Queue using linked list.

UNIT – IV

Searching Techniques: Linear search and Binary Search algorithms.
Sorting Techniques: Bubble Sort, Insertion sort, Selection Sort, Merge Sort, and Quick Sort. Comparison among sorting techniques.

UNIT – V

Trees: Introduction, Binary Trees, Tree Traversals, Threaded Binary Trees, Binary Search Tree, Heap Tree, AVL Tree.
Graphs: Graph Abstract Data Type, Representation of Graph, Graph Traversals -DFS and BFS, Introduction to Spanning Tree.

Suggested Readings:

1. Fundamentals of Data Structures in C, Ellis Horowitz, Sartaj Sahani, Susan Anderson Freed, Computer Science Press.
2. Data Structures, S. Lipschutz, Tata McGraw Hill Education, 4th Edition, 2018.
3. Classic Data Structures, D. Samanta, PHI Learning, 2nd Edition, 2004.
4. Data Structures and Algorithm Analysis In C, Mark A Weiss, Second Edition (2002),Pearson.

Course Code	Course Title						Core/Elective
U23IT607	INTRODUCTION TO WEB PROGRAMMING						Open Elective-II
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
Java Programming	L	T	D	P			
	3	-	-	-	40	60	3

Course Objectives:

Develop ability to

1. Understand the technologies used in Web Programming.
2. Understand the importance of CSS in HTML for styling web pages.
3. Develop dynamic web pages using Java Script.
4. Understand the document structure and schemas and represent data in that format.
5. Understand the basic concepts of PHP for dynamic website designing.

Course Outcomes:

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

1. Develop static web pages using HTML and CSS.
2. Apply and work with various styling techniques.
3. Create dynamic pages using java script.
4. Develop web content publishing for applications and work web services.
5. Analyze dynamic websites using PHP

UNIT I

Introduction to HTML: Origins and Evolution of HTML and XHTML, Basic Syntax, Standard XHTML Document structure, HTML- Basic tags, Form elements and attributes, validation, Frames, HTML 5.0

UNIT II

Cascading Style Sheets: Introduction, Levels of Style sheets, Font properties, List Properties, CSS selectors, CSS BOX Model, Background Images.

UNIT III

Basics of Java Script: Java Script-Object, names, literals, operators and expressions- statements and features- events - windows documents - frames - data types - built-in functions- Browser object model - Verifying forms.

Unit IV

XML and Web Services: The Syntax of XML, XML Document Structure, Document Type Definitions, Name Space, XML Schemas.

Web services: UDDI-WSDL-Java web services – Web resources.

Unit V

Introduction to PHP: Origins and uses, Overview, general syntax, primitive, operations and expressions, control statements. Comparison of PHP and HTML

Suggested Readings:

1. Programming with World Wide Web, Robert W. Sebesta, Eighth Edition, Pearson Education.
2. Web Programming-Building Internet Applications, Chris Bates ,Second Edition, Wiley 2010
3. Java Script-The Complete Reference, Thomas Powell, Fritz Schneider,Third Edition,McGrawHill, 2006
4. Learning PHP, My SQL, Java Script, CSS, HTML 5 - A step by step guide to creating dynamic websites, Robin Nixon, Third Edition, OREILLEY, 2011.