LORDS INSTITUTE OF ENGINEERING AND TECHNOLOGY
(An Autonomous Institution)
DEPARTMENT OF COMPUTER SCIENCE
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	Design and Analysis of Algorithm	3	-	-	3	40	60	3	3
2	U23IT501	PCC	Automata Theory, Languages And Computation	3	1	-	3	40	60	3	4
3	U23CD501	PCC	Software Engineering	3	-	-	3	40	60	3	3
4	U23CM502	PCC	Artificial Intelligence	3	-	-	4	40	60	3	3
5	--	OEC	Open Elective – I	3	-	-	3	40	60	3	3
Practical / Laboratory Course											
6	U23CM5L2	PCC	Artificial Intelligence Lab	-	-	3	3	25	50	3	1.5
7	U23CD5L1	PCC	Software Engineering Lab	-	-	3	3	25	50	3	1.5
8	U23CS5L1	PCC	Design and Analysis of Algorithm Lab	-	-	3	3	25	50	3	1.5
Project											
9	U23CR5P1	PROJ	Internship (During vacation period after 4Sem.)	-	-	2	2	50	-	-	1
Skill Development Course											
10	U23MA5L1	BSC	Aptitude and Reasoning	-	-	2	2	25	50	3	1
Total				**15**	**1**	**13**	**29**	**350**	**500**	**-**	**22.5**

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 **PCC:** Programme Core Courses

BSC: Basic Science Courses **MA**: Mathematics

HSMC: Humanities and Social Sciences **IT:** Information Technology

PCC: Programme Core Courses **PROJ:** Project

MB: Master of Business Administration **CM**: Computer Science and Engineering-AIML

CR: Computer Science

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.

LORDS INSTITUTE OF ENGINEERING AND TECHNOLOGY
(An Autonomous Institution)
DEPARTMENT OF COMPUTER SCIENCE
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	U23CR601	PCC	Compiler Design	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
4	--	PEC	Professional Elective – I	3	-	-	3	40	60	3	3
5	--	OEC	Open Elective – II	3	-	-	3	40	60	3	3
	Practical / Laboratory Course										
6	U23CD6L2	PCC	Computer Networks Lab	-	-	3	3	25	50	3	1.5
7	U23CM6L2	PCC	Machine Learning Lab	-	-	3	3	25	50	3	1.5
8	U23CR6L2	PCC	Web Application Development Lab	-	-	2	2	50	50	3	1.5
	Project										
9	U23CR6P1	PROJ	Mini Project	-	-	6	6	50	50	3	3
Total				**15**	**-**	**14**	**29**	**350**	**450**	**-**	**22**

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   **HSMC:** Humanities and Social Sciences

. **IT:** Information Technology   **PCC:** Programme Core Courses

PE: Professional Elective   **EN:** English

PROJ: Project   **CD**: Computer Science and Engineering-DS

CR: Computer Science

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	DESIGN AND ANALYSIS OF ALGORITHMS						Core
	Contact Hours per Week						
Prerequisite	L	T	D	P	CIE	SEE	Credits
Data Structures	3	-	-	-	40	60	3

Course Objectives:

This course will enable students 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:

On completion of this course, the students are 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 etc. to solve real world problems.
4. Introduce the P and NP classes.
5. Develop solutions for n - Queens problem, Subset – Sum Problem, Assignment problem, Knapsack problem etc.

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: Divide and Conquer: General Method, Binary Search, finding minimum and maximum MergeSort 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 Path

Dynamic programming: The General Method, multistage graph, Warshall’s Algorithm, Floyd’sAlgorithm 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 - Hard and NP, 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 (Chapters 1-5,7,9,11), Pearson Education, Delhi, 2nd Edition, 2007, ISBN: 9780321358288.
2. "Fundamentals of Computer Algorithms", Ellis Horowitz, Sartaj Sahni, Sanguthevar Rajasekaran: (Chapters 1,3-8,10-12), 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 Edu.

<table>
<tr><td><b>Course Code</b></td><td colspan="6"><b>Course Title</b></td><td><b>Core/Elective</b></td></tr>
<tr><td><b>U23IT501</b></td><td colspan="6"><b>AUTOMATA THEORY, LANGUAGES AND COMPUTATION</b></td><td><b>CORE</b></td></tr>
<tr><td><b>Prerequisite</b></td><td colspan="4"><b>Contact Hours Per Week</b></td><td rowspan="2"><b>CIE</b></td><td rowspan="2"><b>SEE</b></td><td rowspan="2"><b>Credits</b></td></tr>
<tr><td rowspan="2"><b>Discrete Mathematics</b></td><td><b>L</b></td><td><b>T</b></td><td><b>D</b></td><td><b>P</b></td></tr>
<tr><td><b>3</b></td><td><b>-</b></td><td><b>-</b></td><td><b>-</b></td><td><b>40</b></td><td><b>60</b></td><td><b>3</b></td></tr>
</table>

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, 2007.
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
U23CD501	**SOFTWARE ENGINEERING**						**Core**
Prerequisite	**Hours Per Week**				**CIE**	**SEE**	**Credits**
	L	**T**	**D**	**P**			
-	**3**	**-**	**-**	**-**	**40**	**60**	**3**

Course Objectives:

1. To introduce the basic concepts of software development- processes from defining a product to shipping andmaintaining that product.
2. To impart knowledge on various phases, methodologies and practices of software development.
3. To understand the importance of testing in software development and study various testing strategies and softwarequality metrics.
4. To impart knowledge on various phases, methodologies and practices of software development
5. To understand the importance of testing in software development, study various testing strategies along with itsrelationship with software quality and metrics

Course Outcomes:

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

1. Describe and compare alternative approaches and techniques used across various phases of the software development lifecycle.
2. Develop a complete software project independently by applying appropriate design principles, tools, and methodologies.
3. Identify and analyze the real-world challenges involved in developing large-scale software systems.
4. Design and construct software architecture independently or in a team by recognizing recurring problems and applying relevant design patterns.
5. Evaluate software product quality using appropriate metrics while addressing practical development challenges.

UNIT-I

Introduction to Software Engineering: The evolving role of software, changing nature of software, software myths. A Generic view of process: Software engineering- a layered technology, a process framework, the capability maturity model integration (CMMI). Process models: The waterfall model, Spiral model and Agile methodology

UNIT – II

Software Requirements: Functional and non-functional requirements, user requirements, system requirements, interface specification, the software requirements document. Requirements engineering process: Feasibility studies, requirements elicitation and analysis, requirements validation, requirements management.

UNIT – III

Design Engineering: Design process and design quality, design concepts, the design model. Creating an architectural design: software architecture, data design, architectural styles and patterns, architectural design, conceptual model of UML, basic structural modeling, class diagrams, sequence diagrams, collaboration diagrams, use case diagrams, component diagrams.

UNIT – IV

Testing Strategies: A strategic approach to software testing, test strategies for conventional software, black-box and white-box testing, validation testing, system testing, the art of debugging. Metrics for Process and Products: Software measurement, metrics for software quality.

UNIT – V

Risk management: Reactive Vs proactive risk strategies, software risks, risk identification, risk projection, risk refinement, RMMM. Quality Management: Quality concepts, software quality assurance, software reviews, formal technical reviews, statistical software quality assurance, software reliability, the ISO 9000 quality standards

Suggested Readings:

1. Software Engineering, A practitioner's Approach- Roger S. Pressman, 8th edition, McGraw Hill International Edition.
2. Software Engineering- Sommerville, 7th edition, Pearson Education.

Course Code	Course Title						Core/Elective
U23CM502	ARTIFICIAL INTELLIGENCE						Core
Prerequisite	Hours Per Week				CIE	SEE	Credits
	L	T	D	P			
Data Structures, Basics of probability	3	1	_	_	40	60	4

Course Objectives:

1.To understand the various characteristics of Intelligent agents

2.To learn the different search strategies in AI

3.To learn to represent knowledge in solving AI problems

4.To understand the different ways of designing software agents

5.To know about the various applications of AI.

Course Outcomes:

At the end of the course, student would 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

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

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

Course Code	Course Title						Core/Elective
U23CM5L2	Artificial Intelligence Lab						Core
Prerequisite	Contact Hours per Week				CIE	SEE	Credits
	L	T	D	P			
-	-	-	-	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. To introduce students to the basic concepts of AI Searching techniques
2. To develop skills for solving practical problems.
3. To gain experience neural networks.

Course Outcomes:

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

1. After learning the AI concepts the student must be able to design and implement AI solutions searching techniques using AI.
2. Able to know about facts of querying.
3. Be capable of confidently applying tree mechanism using AI with Neural Network
4. Be capable of performing experiments in Machine Learning using real-world data.·
5. Be capable to implement classifiers and Regression algorithm

List of Experiments:

1. Write a Program to Implement Breadth First Search
2. Write a Program to Implement Depth First Search
3. Write a Program to Implement Tic-Tac-Toe game
4. Write a Program to Implement 8-Puzzle problem
5. Write a Program to Implement Water-Jug problem
6. Write a Program to Implement Travelling Salesman Problem
7. Write a Program to Implement Tower of Hanoi
8. Write a Program to Implement Monkey Banana Problem
9. Write a Program to Implement Missionaries-Cannibals Problems
10. Write a Program to Implement N-Queens Problem using Python.
11. Write a program to train and validate the following classifiers for given data (scikit-learn):

a) Decision Tree

b) Multi-layer Feed Forward neural network

c) Implementation of Gaussian Naive Bayes classifier using scikit-learn (Any two classifiers).

12. Write a program to Implementation of Linear Regression using python(Any Two Algorithm).

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

<table>
<tr><td>Course Code</td><td colspan="6"><b>Course Title</b></td><td>Core/Elective</td></tr>
<tr><td><b>U23CD5L1</b></td><td colspan="6"><b>SOFTWARE ENGINEERING LAB</b></td><td><b>Core</b></td></tr>
<tr><td rowspan="2"><b>Prerequisite</b></td><td colspan="4"><b>Hours Per Week</b></td><td rowspan="2"><b>CIE</b></td><td rowspan="2"><b>SEE</b></td><td rowspan="2"><b>Credits</b></td></tr>
<tr><td><b>L</b></td><td><b>T</b></td><td><b>D</b></td><td><b>P</b></td></tr>
<tr><td></td><td><b>-</b></td><td><b>–</b></td><td><b>–</b></td><td><b>3</b></td><td><b>25</b></td><td><b>50</b></td><td><b>1.5</b></td></tr>
</table>

Course Objectives:

1. To identify Project Scope, Objectives and infrastructure
2. To understand software engineering methodologies for project development.
3. To capture the requirements specification for an intended software system
4. To draw the UML diagrams for the given specification
5. To map the design properly to code

Course Outcomes:

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

1. Recall the basic principles of Software Engineering and UML modeling.
2. Interpret and explain use-case diagrams, class diagrams, and activity diagrams for software requirements.
3. Construct various UML diagrams (use case, class, sequence, collaboration, activity, state, component, deployment).
4. Analyze software requirements to select appropriate modeling techniques and tools.
5. Evaluate the consistency and correctness of UML diagrams with respect to given requirements.

List of Experiments

1. Understanding UML diagrams
2. Understanding the software to implement UML diagrams
3. Design the **Use Case Diagram** highlighting student and librarian interactions.
4. Draw the **Class and Object Diagrams** for the system structure and a sample book issue transaction.
5. Create the **Sequence Diagram** and **Collaboration Diagram** for the book issue scenario.
6. Create **State chart** and **Activity Diagrams** showing workflows for ATM transaction and LMS book issue process.
7. Develop the **Component and Deployment Diagram** indicating physical deployment on hardware infrastructure.
8. Design an UML diagrams for Exam Registration Process
9. Design an UML diagrams for credit card Processing

Suggested Readings:

1. The Unified Modeling Language User Guide Grady Booch, James Rambaugh, Ivar Jacobson, Pearson Education.
2. UML Distilled Third Edition, Martin Fowler.

Course code	**Course Title**						Core/Elective
U24CS5L1	**Design and Analysis of Algorithms Lab**						**Core**
Pre-requisites	Contact Hours Per Week				CIE	SEE	Credits
DSA	**L**	**T**	**D**	**P**			
	-	-	-				1.5

Course Objectives:

The objectives of this course are:

1. Understand problems by applying appropriate algorithms.
2. Analyze the efficiency of various algorithms.
3. Apply techniques of stacks and queues to solve problems.
4. Solve a program in many ways using different techniques.
5. Identify and evaluate complex problems using principles of mathematics and engineering science.

Course Outcomes

At the end of this Course, the student will able to:

1. Solve problems by applying appropriate algorithms.
2. Analyze the efficiency of various algorithms.
3. Apply techniques of stacks and queues to solve problems.
4. Develop a program that can be solved in many ways using different techniques.
5. Identify and evaluate complex problems using principles of mathematics and engineering science.

Compare all the program implementations empirically with the brute force technique. Almost all the experiments from syllabus are listed, the department can decide and choose any subset of the listed experiments.

1. The student should be able to code program implementing and comparing all the learned sorting algorithms such as Bubble, quick, Heap, merge, insertion, selection sort in terms of number of swaps, number of comparisons and should be able to write report justifying which technique of sorting is useful in different scenarios. The student should also sort objects, not just numerical values.
2. The student should be able to code Linear search and binary search and justify where which learning technique is to be used. The searching is done also on objects, not just numbers alone.
3. A simple program demonstrating the Los Vegas, compare it, with the deterministic approach
4. A simple program demonstrating the Monte Carlo, compare it, with the deterministic approach
5. Implement randomized quick sort, compare it with the deterministic approach.
6. Solve the max-min using divide and conquer and compare it with the brute force technique
7. Implement the Karatsuba method and compare it with the brute force technique
8. Obtain the median of given list using the decrease and conquer technique
9. Solve the currency change problem with minimization and maximization.
10. Implement knapsack, job sequence with deadline by greedy technique
11. Implement Huffman coding and single source shortest path algorithm

12. Obtain the Fibonacci numbers using Dynamic programming in both (memorization and tabulation)
13. Implement Optimal binary search tree, 0/1 knapsack problem using Dynamic programming strategy
14. Implement All pairs shortest path problem, Traveling salesperson problem, multi-stage graph
15. Implement Floyd's Algorithm All pairs shortest algorithm, Reliability design
16. Implement graph coloring and Hamiltonian cycle using backtracking
17. Implement Traveling salesperson problem, 0/1 knapsack problem using branch and bound
18. A simple program demonstrating approximate algorithms, compare it with conventional method.

Course Code	**Course Title**						**Core/ Elective**
U23CR601	**COMPILER DESIGN**						**CORE**
Prerequisite	**Contact Hours Per Week**				**CIE**	**SEE**	**Credits**
	L	**T**	**D**	**P**			
-	**3**	**1**	**-**	**-**	**30**	**70**	**3**

Course Objectives

1. To understand and list the different stages in the process of compilation.
2. Identify different methods of lexical analysis
3. Design top-down and bottom-up parsers
4. Identify synthesized and inherited attributes
5. Develop syntax directed translation schemes
6. Develop algorithms to generate code for a target machine

Course Outcomes

1. Upon completion of the course, the students will be able to:
2. For a given grammar specification, develop the lexical analyzer.
3. For a given parser specification, design top-down and bottom-up parsers.
4. Develop syntax directed translation schemes.
5. Develop algorithms to generate code for target machine.

UNIT-I

Introduction: The Structure of a Compiler, Phases of Compilation, The Translation Process, Major Data Structures in a Compiler, Bootstrapping and Porting.

Lexical Analysis (Scanner): The Role of the Lexical Analyzer, Input Buffering, Specification of Tokens, Recognition of Tokens, The Lexical Analyzer Generator Lex.

UNIT-II

Syntax Analysis (Parser): The Role of the Parser, Syntax Error Handling and Recovery, Top-Down Parsing, Bottom-Up Parsing, Simple LR Parsing, More Powerful LR Parsing, Using Ambiguous Grammars, Parser Generator Yaac.

UNIT-III

Syntax-Directed Translation: Syntax-Directed Definitions, Evaluation Orders for SDD's Applications of Syntax-Directed Translation.

Symbol Table: Structure, Operations, Implementation and Management.

UNIT-IV

Intermediate Code Generation: Variants of Syntax Trees, Three-Address Code, Types and Declarations, Translation of Expressions, Type Checking, Control Flow, Backpatching, Switch-statements, Intermediate Code for Procedures.

Run-time environment: Storage Organization, Stack Allocation of Space, Access to Nonlocal Data on the Stack, Parameter passing, Heap Management and Garbage Collection.

UNIT-V

Code Generation: Issues in the Design of a Code Generator, The Target Language, Addresses in the Target Code, Basic Blocks and Flow graphs, Optimization of Basic Blocks, Peephole Optimization, Register Allocation and Assignment.

Machine-Independent Optimizations: The Principal Sources of Optimizations, Introduction to Data-Flow Analysis.

Suggested Books:

1. Alfred V. Aho, Monica S. Lam, Ravi Sethi, & Jeffrey D. Ullman , Compilers s:*Principles, Techniques and Tool***s**, *2nd Edition, Pearson Education, 2006.*
2. Kenneth C. Louden**,** Compiler Construction: Principles and Practice, Thomson Learning Inc., 1997.
3. P.Trembley and P.S.Sorenson, The Theory and Practice of Compiler Writing, TMH- 1985.

Course Code	Course Title						Core / Elective
U23CD602	COMPUTER NETWORKS						Professional Elective
Prerequisite	Contact Hours per Week				CIE	SEE	Credits
	L	T	D	P			
-	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
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.

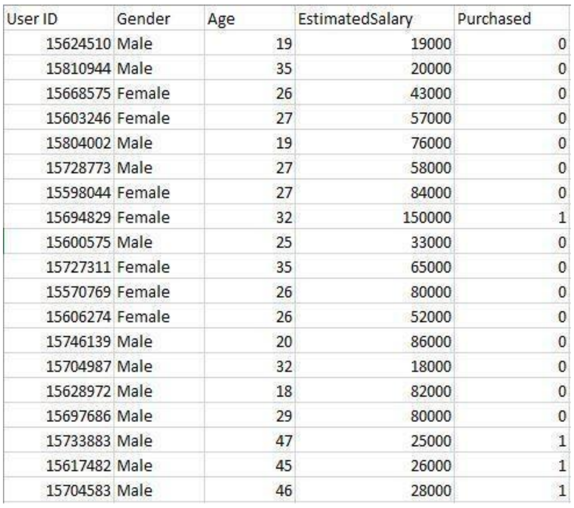
2. Using python write a NumPy program to compute the
   a) Expected Value
   b) Mean
   c) Standard deviation
   d) Variance
   e) Covariance
   f) Covariance Matrix of two given arrays.

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.

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

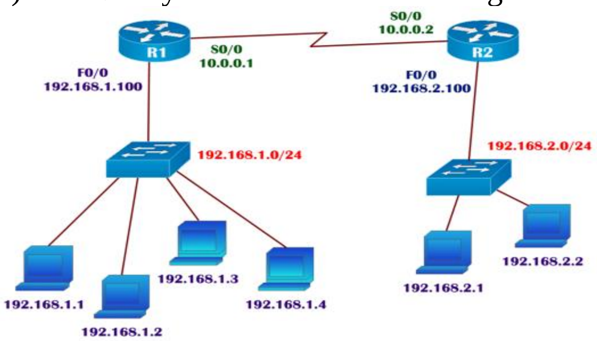
Course Objectives:

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:

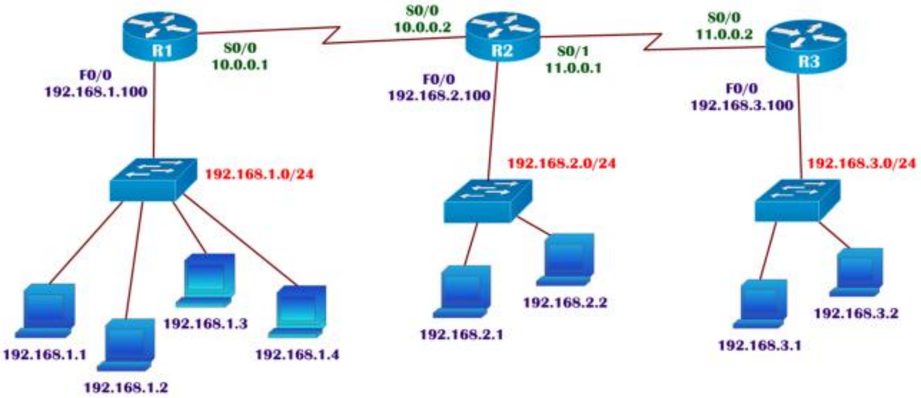
1. Explain & design the various reference models and networks.
2. Identify the different types of network devices and Multiple Access Protocols.
3. Use IP addressing Scheme and to interconnect various networks and Routing mechanism
4. Explain transport layer protocols: TCP, UDP.
5. Explain and use various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

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 c) Verify the interface status using commands



Sr No	Experiment No.	NAME OF THE EXPERIMENT
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**

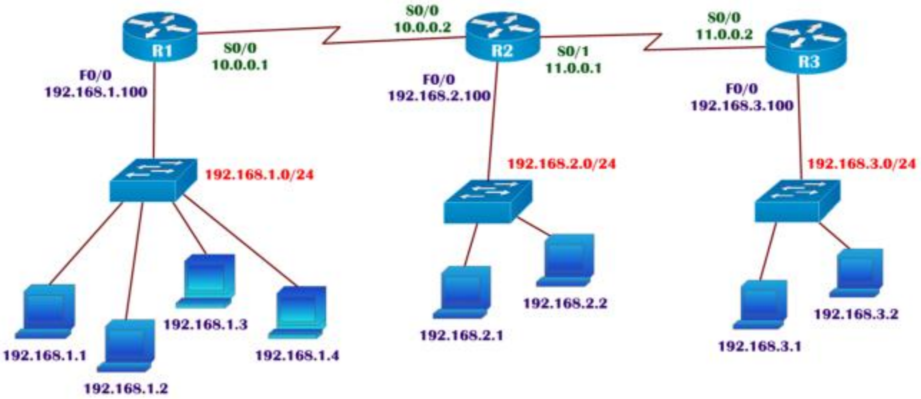
a) Design the topology

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**

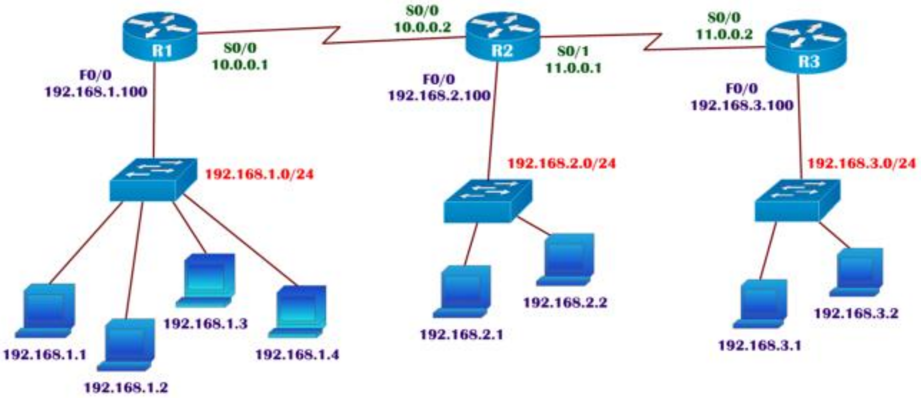
a) Design the topology

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
U23CR6L2	WEB APPLICATION DEVELOPMENT LAB						CORE
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
Java	L	T	D	P			
	-	-	-	2	50	-	1

Course Objectives

Develop ability to

1. Develop web pages using HTML tags and perform validation using scripting
2. Implement various types of styling using CSS and transform data into various forms
3. Understand, analyze and create XML documents and XML Schema
4. Implement applications using JQuery and AngularJS
5. Understand and implement the concepts of MEAN Stack and SMACK stack Course

Course Outcomes

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

1. Design Web pages and perform form validation using HTML5.0 in built functions.
2. Apply Styles to the web content using CSS.
3. Create and process web publishing content using XML and JSON.
4. Use JQuery to perform client-side Dynamics.
5. Create single page applications(Front End)using AngularJS. And design Big data applications using Mean stack or SMACK stack Frameworks.

List of Experiments

1. Implement HTML:
   i. Basic HTML Tags
   ii. Design a Class Time Table using Table Tag & its Attributes
   iii. Implement FRAMES
2. Design the following static web pages required for an online book store web site.
   i) HOME PAGE: The static home page must contain three frames.
   ii) LOGIN PAGE
   iii) CATOLOGUE PAGE: The catalogue page should contain the details of all the books available in the web site in a table.
   iii) REGISTRATION PAGE
3. Create a web page using
   i) Inline CSS,
   ii) Internal CSS
   iii) External CSS
4. Write JavaScript to validate the following fields of the Registration page.
   1. First Name (Name should contains alphabets and the length should not be less than 6 characters).
   2. Password (Password should not be less than 6 characters length).
   3. E-mail id (should not contain any invalid and must follow the standard pattern name@domain.com)
   4. Mobile Number (Phone number should contain 10 digits only).
   5. Last Name and Address (should not be Empty).
   document
5. Display the various forms of XML document
   i) Raw XML ii) XML using CSS
6. Write a program to create a website using HTML, CSS and JavaScript?
7. Write a program to create a voting application using React JS.

8. Write a program to create and Build a Password Strength Check using Jquery.

9. Create a NodeJS server that serves static HTML and CSS files to the user without using Express.

10. Create a NodeJS server using Express that stores data from a form as a JSON file and displays it in another page. The redirect page should be prepared using Handlebars.

Suggested Readings:

1. Web Technologies, Oxford University Press, UttamK Roy, 1st Edition,2010.
2. Java Script: The Definitive Guide, David Flanagan O’Reilly, 6th Edition

6th Sem	**7th Sem**		**8th Sem**	
PE-I	**PE-II**	**PE-III**	**PE-IV**	**PE-V**
Natural Language Processing	Computer Vision	Image Processing	Computational Intelligence	Generative AI
Software Quality & Testing	Agile Methodologies	DevOps	Software Project Management	Human Computer Interaction
Data Mining	Data Warehousing	Statistical Learning	Data Visualization	Cloud-Based Data Processing
Mobile Computing	Wireless Sensor Networks	Network Protocols and Simulation	5G & Beyond	Software Defined Networking

Course Code	Course Title	Core/Elective
U21IT802	NATURAL LANGUAGE PROCESSING	Professional Elective-IV

Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
AI/ML	L	T	D	P			
	3	-	-	-	40	60	3

Course Objectives

Develop ability to

1 Learn the fundamentals of natural language processing.

2. Understand the various text processing techniques in NLP.

3. Understand the role Text Classification Deep Learning for Text Classification techniques of NLP

4. Use Modelling for Case Studies and apply the NLP techniques to IR applications.

5. Learn Ecommerce and the Applications used for NLP.

Course Outcomes

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

1. Understand the basic concepts of Natural language processing pipeline and applications of NLP.
2. Illustrate various text representation techniques in NLP.
3. Analyse text classification techniques and deep learning basics to process natural language text.
4. Explore text summarization methods and example systems.
5. Demonstrate levels of NLP for several case studies.

UNIT – I

NLP: A Primer, NLP in the Real World, NLP Tasks, NLP Levels, What Is Language? Building Blocks of Language, Why Is NLP Challenging? Machine Learning and Overview Approaches to NLP, Heuristics- Based, Machine Learning, Deep Learning for NLP. NLP Pipeline: Data Acquisition, Pre-Processing Preliminaries Frequent Steps, Advanced Processing Feature Engineering Classical NLP/ML Pipeline.

UNIT – II

Text Representation Vector Space Models Basic Vectorization Approaches, One-Hot Encoding Bag of Words, Bag of N-Grams, TF-IDF, Distributed Representations, Word Embedding

UNIT- III

Text Classification Applications One Pipeline, Many Classifiers, Using Neural Embeddings in Text Classification Deep Learning for Text Classification Interpreting Text Classification Models. Deep Learning for Text Classification CNNs for Text Classification.

UNIT – IV

Topic Modelling Text Summarization, Use Cases Setting Up a Summarizer: An Example Recommender Systems for Textual Data Machine Translation Question-Answering Systems, Social Media, E-Commerce and Retail.

UNIT - V

Case Study on NLP Pipeline, Text Classification: Ticketing, Ecommerce, Social media, health care, Recommender systems and other applications of NLP, Language Model Adaptation, Types of Language Models.

Suggested Readings:

1. Sowmya Vajjala, Bodhisattwa Majumder, Anuj Gupta & Harshit Surana “Practical Natural Language Processing: A Comprehensive Guide to Building Real world NLP Systems”, O‟Reilly Media, Inc., 1st Edition, 2020. 2. James Allen, “Natural Language Understanding”, Bejamin Cummings, 2nd edition, 1995.

2. Tanveer Siddiqui, U.S. Tiwary, “Natural Language Processing and Information Retrieval”, Oxford University Press, 2008.

Course Code	Course Title						Core/Elective
U21IT802	Software Quality and Testing						Professional Elective-IV
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
AI/ML	L	T	D	P			
	3	-	-	-	40	60	3

Course Objectives

Develop ability to

1.To understand the importance of software quality and the challenges involved in achieving high-quality software.

2.To learn how to integrate quality assurance activities throughout the software development life cycle.

3.To gain knowledge about various software quality metrics and international quality standards like ISO, CMM, and Six Sigma.

4.To understand the complete software testing life cycle including test planning, test design, test execution, and defect reporting.

5.To explore tools and techniques for testing different types of software applications including web-based, client-server, and security-critical systems

Course Outcomes

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

1.Student will be able to explain the importance of software quality and identify different software quality factors.

2.Student will be able to apply quality assurance techniques at various stages of the software development life cycle.

3.Student will be able to evaluate software quality using quality metrics and follow standard models like ISO, CMMI, and Six Sigma.

4.Student will be able to develop and execute effective software test plans using appropriate testing techniques and methodologies.

5.Student will be able to select and use suitable automated testing tools to test software applications in different environments.

UNIT – I

The Software Quality Challenge, Introduction Software Quality Factors, The Components of the Software Quality Assurance System – Overview, Development and Quality Plans.

UNIT – II

Integrating Quality Activities in the Project Life Cycle, Assuring the Quality of Software Maintenance Components, CASE Tools and their effect on Software Quality, Procedure and Work Instructions, Supporting Quality Devices, Configuration Management, Documentation Control, Project Progress Control.

UNIT – III

Software Quality Metrics, Costs of Software Quality, Quality Management Standards – ISO 9000 and Companion ISO Standards, CMM, CMMI, PCMM, Malcom Balridge, 3 Sigma, 6 Sigma, SQA Project Process Standards – IEEE Software Engineering Standards.

UNIT – IV

Building a Software Testing Strategy, establishing a Software Testing Methodology, Determining Your Software Testing Techniques, eleven – Step Software Testing Process Overview, Assess Project Management Development Estimate and Status, Develop Test Plan, Requirements Phase Testing, Design Phase Testing, Program Phase Testing, Execute Test and Record Results, Acceptance Test, Report Test Results, Test Software Changes, Evaluate Test Effectiveness.

UNIT – V
Testing Client / Server Systems, Testing the Adequacy of System Documentation, Testing Web-based Systems, Testing Off – the – Shelf Software, testing in a Multiplatform Environment, Testing Security, testing a Data Warehouse, Creating Test Documentation, Software Testing Tools, Taxonomy of Testing Tools, Methodology to Evaluate Automated Testing Tools, Load Runner, Win Runner and Rational Testing Tools, Java Testing Tools, JMetra, JUNIT and Cactus.

Suggested Readings:

1. Daniel Galin, Software Quality Assurance–From Theory to Implementation, Pearson Education.2004
2. Mordechai Ben Menachem / Garry S. Marliss, Software Quality–Producing Practical, Consistent
Software, BS Publications, 2014
3. William E. Perry, Effective Methods for Software Testing, 2nd Edition, Wiley.
4. Srinivasan Desikan, Gopalaswamy Ramesh, Software Testing, Principles and Practices, 2006. Pearson Education.
5. K.V.K.K. Prasad, Software Testing Tool, Wiley Publishers

Course Code	Course Title						Core/Elective
U21IT802	Data Mining						Professional Elective-IV
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
AI/ML	L	T	D	P			
	3	-	-	-	40	60	3

Course Objectives

Develop ability to

1.To understand the fundamental concepts of data mining, its purpose, scope, and challenges.
2.To study data preprocessing techniques including data types, statistical descriptions, and similarity measures.
3.To explore various methods for discovering patterns such as frequent itemsets, associations, and correlations.
4.To learn classification and clustering techniques for analyzing large datasets.
5.To examine current trends, research areas, and ethical considerations in data mining.

Course Outcomes

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

1.Student will be able to explain the concepts, applications, and challenges of data mining.
2.Student will be able to apply data preprocessing and similarity measurement techniques for data analysis.
3.Student will be able to discover and evaluate frequent patterns, associations, and correlations in datasets.
4.Student will be able to build and assess classification and clustering models using appropriate data mining algorithms.
5.Student will be able to discuss the recent trends, societal impacts, and research frontiers in data mining.

UNIT-I

Introduction: Why Data Mining, What is Data Mining, What kinds of data can be mined? What kinds of patterns can be mined? Which technologies are used? Which kinds of applications are Targeted? Major issues in Data Mining. Getting to know your data: Data objects and attributed types. Basic statistical descriptions of data. Data visualization, Measuring data similarity and dissimilarity.

UNIT-II

Mining frequent patterns, Associations and correlations: Basic concepts and methods, Frequent Item set Mining Methods, which patterns are interesting? Pattern evaluation methods.

UNIT-III

Classification: Basic concepts, Decision tree induction, Bayes classification methods, Advance methods, Bayesian Belief Network, Classification by back propagation, Support vector machine.

UNIT-IV

Cluster Analysis: Concepts and Methods: Cluster Analysis, Partitioning Methods, Hierarchical Methods, Density-Based Methods, Grid-Based Methods, Evaluation of clustering.

UNIT-V

Data Mining Trends and Research Frontiers: Mining Complex Data Types, Other Methodologies of Data Mining, Data Mining Applications, Data Mining and Society, Data Mining trends.

Suggested Readings:

1. Jiawei Han, Micheline Kamber, Jin Pei, Data Mining: Concepts & Techniques, 3rd Edition., Morgon Koffman ,2011
2. Vikram Pudi, P. Radha Krishna, Data Mining, Oxford University Press, 1st Edition, 2009.
3. Pang-Ning Tan, Michael Steinbach, Vipin Kumar, Introduction to Data Mining, Pearson Education, 2008.

Course Code	Course Title						Core/Elective
U21IT802	Mobile Computing						Professional Elective-IV
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
AI/ML	L	T	D	P			
	3	-	-	-	40	60	3

Course Objectives

Develop ability to

1.To introduce the basic concepts, applications, and architecture of mobile computing systems.
2.To understand how GSM and wireless communication systems work, including MAC techniques.
3.To explore how mobile IP and mobile transport layer protocols manage mobility and data transfer.
4.To study the design and working of Mobile Ad hoc Networks (MANETs) and their routing protocols.
5.To learn about mobile communication tools and technologies like WAP, Bluetooth, and J2ME.

Course Outcomes

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

1.Student will be able to describe mobile computing concepts and explain GSM system architecture.
2.Student will be able to explain how mobile devices connect to networks using Mobile IP and DHCP.
3.Student will be able to compare different transport layer protocols used in mobile environments.
4.Student will be able to understand routing protocols used in Mobile Ad hoc Networks (MANETs).
5.Student will be able to identify and use mobile communication technologies like WAP, Bluetooth, and J2ME.

UNIT- I

Introduction to MC, Applications, limitations, and architecture. GSM : Mobile services, System architecture, Radio interface, Protocols, Localization and calling, Handover, Security, and New data services.(Wireless) Medium Access Control : Motivation for a specialized MAC (Hidden and exposed terminals, Near and far terminals), SDMA, FDMA, TDMA, CDMA.

UNIT-II

Mobile Network Layer: Mobile IP (Goals, assumptions, entities and terminology, IP packet delivery, agent advertisement and discovery, registration, tunneling and encapsulation, optimizations), Dynamic Host Configuration Protocol (DHCP).

UNIT -III

Mobile Transport Layer : Traditional TCP, Indirect TCP, Snooping TCP, Mobile TCP, Fast retransmit/fast recovery, Transmission /time-out freezing, Selective retransmission, Transaction oriented TCP.

UNIT- IV

Mobile Ad hoc Networks (MANETs): Routing, Destination sequence, distance vector Dynamic source routing alternative metrics, overview Adhoc routing protocols.

UNIT -V

Protocols and Tools: Wireless Application Protocol-WAP. (Introduction, protocol architecture, and treatment of protocols of all layers), Bluetooth User scenarios, Architecture, security, link management) and J2ME.

Suggested Readings:

1. Jochen Schiller, Mobile Communications, 2nd edition, Addison-Wesley, 2004.
2. Asoke K Talukder, Mobile Computing , 2nd Edition, McGraw Education
3. Reza Behravanfar, Mobile Computing Principles, Designing and Developing Mobile Applications with UML and XM, ISBN: 0521817331, Cambridge University Press, October 2004,
4. Adelstein, Frank, Gupta, Sandeep KS, Richard III, Golden, Schwiebert, Loren, Fundamentals of Mobile and Pervasive Computing, ISBN: 0071412379, McGraw-Hill Professional, 2005.
5. Hansmann, Merk, Nicklous, Stober, Principles of Mobile Computing, Springer, second edition, 2003.
6. Martyn Mallick, Mobile and Wireless Design Essentials, Wiley DreamTech, 2003.