UNIVERSITY OF PETROLEUM & ENERGY STUDIES

(ISO 9001:2008 Certified)

MCA (Master of Computer Application) with specialization in Artificial Intelligence

(VERSION 1.0)

w.e.f. 2021

UPES Campus
"Energy Acres"
P.O Bidholi via Prem Nagar, Bidholi
Dehradun – 248007
(Uttarakhand)

Tel : + 91-135-2776053/54
Fax: + 91-135-2776090
URL: www.upes.ac.in

INTELLECTUAL PROPERTY RIGHTS

UNIVERSITY OF PETROLEUM & ENERGY STUDIES

MCA (Master of Computer Application) with specialization in Artificial Intelligence

SEMESTER I		
Subject Code	Subject	Credits
CSAI 7011	Python Programming	3
CSEG 7009	Web Technologies Through PHP	3
CSEG 7010	Software Engineering and Project Mgmt	3
HRES 7018	Business Communication and Ethics	3
	Data Structures	4
	Domain Elective-1	3
CSAI 7111	Python Programming Lab	1
CSEG 7109	Web Technologies Through PHP Lab	1
	Data Structures Lab	2
		1
	Total	24

SEMESTER II		
Subject Code	Subject	Credits
CSEG 7011	Data Base Management Systems	3
CSEG 7012	Operating Systems	3
CSEG 7013	Object-Oriented Analysis and Design Using UML	3
UCIE 7004	Venture Ideation and Entrepreneurship	1
	Domain Elective-2	3
	Domain Elective 3	3
CSEG 7014	Java Programming	3
CSEG 7111	Data Base Management Systems Lab	1
	Domain Elective-2 Lab	2
CSEG 7114	Java Programming Lab	1
	Technical Seminar	1
	Total	**24**

	AI
Semester-I	**Domain Elective-1**
	Problem Domains of AI *[5]
Semester-II	**Domain Elective-2**
	Soft Computing
Semester-II	**Domain Elective-3**
	Knowledge Engineering and Expert Systems
Semester-III	**Domain Elective-4**
	Machine Learning - Using Data for Artificial Intelligence
Semester-III	**Domain Elective-5**
	Deep learning and ANN
Semester-III	**Domain Elective-6**
	Natural language Processing
	Game Theory and Heuristics
	Image Processing and Machine Vision
	Information retrieval
	Data Visualization
	Biometric processing
	Business Analytics and Optimization
	Agent based Intelligent Systems

A. PROGRAM OUTCOMES (POs) for Master of Computer Application

A1. PROGRAM OUTCOMES

PO1Computational Knowledge: Apply knowledge of computing fundamentals and domain knowledge.

PO2 Problem Analysis: Identify, formulate and solve complex computing problems reaching substantiated conclusions.

PO3 Development of Solutions: Design and evaluate solutions for complex computing problems with appropriate consideration.

PO4 Investigations of complex Computing problems: Use research-based knowledge and research methods for analysis and

interpretation of data, and synthesis of the information to provide valid conclusions

PO5 Modern Tool Usage: Create, identify and apply appropriate techniques, resources, and modern computing tools to complex

computing activities.

PO6 Professional Ethics: Understand and commit to professional ethics and cyber regulations for professional computing practices.

PO7 Life-long Learning: Identify the need and have the ability, to engage in independent learning as a computing professional.

PO8 Communication Efficacy: Communicate effectively with the computing community, and with society.

PO9 Individual and Team Work: Function effectively in diverse teams and in multidisciplinary environments.

PO10 Innovation and Entrepreneurship: Identify a timely opportunity and using innovation to pursue that opportunity.

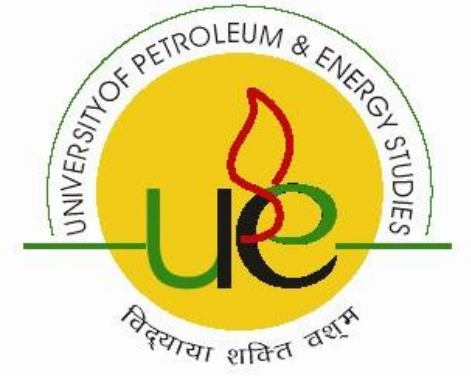
P11 Research Skill - Extract information through literature survey and experiments, apply appropriate research methodologies, techniques and tools, design, conduct experiments, analyze and interpret data, contribute individually/in group(s) to the development of scientific/technological knowledge in one or more domains of computer applications.

P12 Independent and Reflective Learning - Observe and examine critically the outcomes of one's actions and make corrective measures subsequently, and learn from mistakes without depending on external feedback.

A2 PROGRAM SPECIFIC OUTCOMES

PSO1. Apply the knowledge of AI to find solutions for real-life application.

PSO2. Domain Specific Learning and Contribution: Students will be able to design hardware or software solutions in AI, Imaging, Analytics and Security Domains Computer Science basics. Able to contribute in digital and technical transformation of society through humanized product development.



SEMESTER I

	Python Programming	L	T	P	C
Version 1.0		3	0	0	3
Pre-requisites/Exposure	--				
Co-requisites	--				

Course Objectives

1. To learn the basics of python programming.
2. To develop programming skills in python.
3. To understand the intermediate knowledge about various data structures in python language.

Course Outcomes

On completion of this course, the students will be able to

CO1. Know the basic python programming concepts, data structures & regular expressions.
CO2. Discuss file handling operations and understand OOPS concepts using python.
CO3. Develop modules and implement web development framework.
CO4. Discuss the role of python in advance technology.

Catalog Description

Python is a programming language with a simple syntax, and a powerful set of libraries. It is an interpreted language, with a rich programming environment, including a robust debugger and profiler. While it is easy for beginners to learn, it is widely used in many scientific areas for data understanding and exploration. It covers data types, control flow, object-oriented programming, and graphical user interface-driven applications. It also discusses text processing, simple graphics creation and image manipulation, HTML and web programming, and genomics. It enables various web applications concepts.

Course Content

Unit 1.

a) An Introduction to Python

Introduction, A Brief History of Python, Python Versions, Installing Python, Environment Variables, Executing Python from the Command Line, IDLE, Editing Python Files, Python Documentation, Getting Help, Dynamic Types, Python Reserved Words, Naming Conventions

b) Basic Python Syntax

Basic Syntax, Comments, String Values, String Methods, The format Method, String Operators, Numeric Data Types, Conversion Functions, Simple Output, Simple Input, The % Method, The print Function

c) Language Components

Indenting Requirements, the if Statement, Relational and Logical Operators, Bit Wise Operators, the while Loop, break and continue, the for Loop.

Unit 2.

a) Collections

Introduction, Lists, Tuples, Sets, Dictionaries, Sorting Dictionaries, Copying Collections, Summary

b) Functions

Introduction, Defining Your Own Functions, Parameters, Function Documentation, Keyword and Optional Parameters, Passing Collections to a Function, Variable Number of Arguments, Scope, Functions - "First Class Citizens", Passing Functions to a Function, map, filter, Mapping Functions in a Dictionary, Lambda, Inner Functions, Closures

c) Modules

Modules, Standard Modules – sys, math, time, The dir Function

Unit 3.

a) Exceptions

Errors, Runtime Errors, The Exception Model, Exception Hierarchy, Handling Multiple Exceptions, raise, assert.

b) Input and Output

Introduction, Data Streams, Creating Your Own Data Streams, Access Modes, Writing Data to a File, Reading Data from a File, Additional File Methods, Using Pipes as Data Streams, Handling IO Exceptions, Working with Directories, Metadata, The pickle Module

Unit 4.

a) Classes in Python

Classes in Python, Principles of Object Orientation, Creating Classes, Instance Methods, File Organization, Special Methods, Class Variables, Inheritance, Polymorphism, Type Identification, Custom Exception Classes

b) Regular Expressions

Introduction, Simple Character Matches, Special Characters, Character Classes, Quantifiers, The Dot Character, Greedy Matches, Grouping, Matching at Beginning or End, Match Objects, Substituting, splitting a String, Compiling Regular Expressions, Flags.

Unit 5.

a) Data Structures

List Comprehensions, Nested List Comprehensions, Dictionary Comprehensions, Dictionaries with Compound Values, Processing Lists in Parallel, Specialized Sorts, Time Functionality, Generators.

b) Writing GUIs in Python

Introduction, Components and Events, An Example GUI, The Tk Widget, Button Widgets, Entry Widgets, Text Widgets, Checkbutton Widgets, Radiobutton Widgets, Listbox Widgets, Frame Widgets, Menu Widgets, Toplevel Widgets, Dialogs.

Unit 6.

a) The OS Module

The Environment, creating a Process, Listing Files, Other Process Methods, File Information (Metadata), Working with Directories.

b) Numerical Analysis & Plotting

Numpy – Overview, Setup, Datatypes, Basic Operators, Indexing, Broadcasting, Matrix Operators.

Matplotlib-Overview, Setup, Basic plots, Customizing plots, Subplots, 3D plots.

c) Data Processing with Pandas

Pandas – Overview, Setup, Data Structures, Indexing & Selecting Data, groupby Operations, Reshaping data.

Text Book

Programming with Python (IBM ICE Publications 2018 Edition).

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination
Examination Scheme:

Components	MSE	Quiz/Assignment/ etc.	ESE
Weightage (%)	20%	30%	50%

Relationship between the Course Outcomes (COs), Program Outcomes (POs) and Program Specific Outcomes (PSOs):

Course Outcomes	PO1	PO2	PO3	PO 4	PO 5	PO6	PO 7	PO8	PO9	PO 10	PSO 1	PSO2
CO1												
CO2												
CO3												
CO4												
Average												

1=Weak 2=Moderate 3=Strong

	Web Technologies Through PHP	L	T	P	C
Version 1.0		3	0	0	3
Pre-requisites/Exposure	Basic knowledge of web programming				
Co-requisites	--				

Course Objectives

The Course objective is to provide overview on Web technologies and Server Side programming. Upon the completion of this course, the students will be having the knowledge on Web technologies, and will be able to develop the Dynamic Web Programs using PHP and databases.

Course Outcomes

On completion of this course, the students will be able to
CO1. Learn markup languages HTML and scripting languages JavaScript
CO2. Identify the basic PHP programming structures
CO3. Develop real world object-oriented concepts using PHP
CO4. Develop interactive web based program with database connectivity

Catalog Description

Nowadays, the use of web applications are enormous by the individuals and organizations. Companies are using web-based solutions for their customers. Companies are also using CRM software to increase the loyalty of their consumers, and for taking feedbacks from their customers. Banks are using web based solutions to facilitate their customer to do transections, and access required information from anywhere in the world. The use of social and professional networking web sites are increasing day-by-day. For the development of web-based application, companies are recruiting smart-working employees with knowledge of server-side scripting language like PHP and web technologies. In this course, students will be able to understand the different web technologies. They would learn how to develop real world web-based applications using PHP programming. They will also learn how to develop the projects as a team.

Course Content

Unit 1: HTML and JavaScript (9)

What is HTML, Basic Structure of HTML Page, Basic Tags, Lists, Tables, Images, Forms and Frames. Dynamic HTML with Java Script: Data validation, Opening a new window, Messages and Confirmations, The status bar, writing to a different frame, Rollover buttons, Moving images, multiple pages in a single download, a text-only menu system, Floating logos.

Introduction, Usage of variables, operations, control structures, looping structures, predefined keywords, arrays and functions, objects. Exception Handling, Events and Event Handling and Validations.

Unit 2: PHP Basics, Controls Structures and Functions in PHP (9)

Introduction to PHP, Installation, Basic Syntax of PHP, PHP statement terminator and case insensitivity, Embedding PHP in HTML, Comments, Variables, Assigning value to a variable, Constants, Managing Variables.

Operators, String Manipulation, Conditional Control Structures, Looping Control Structures, Break and Continue, Functions, Understanding variable scope, Built-in functions in PHP.

Unit 3: Arrays, PHP File Handling, Class and Objects, Exception Handling (9)

Array in PHP, Creating and accessing array elements, Iterating Array with Numeric index, Multidimensional Array, Accessing elements of a Multidimensional Array, Iterating Multidimensional Array. Introduction, File Operations, Using PHP with HTML Forms.

Defining Class and Object in PHP, Usage of $this variable, Constructor, Constructor with Parameters. Introduction to Exception, Exception Handling mechanisms, Creating Custom Exceptions, Multiple Catch Blocks, Exception Propagation, Error Handling in PHP.

Unit 4: PHP with Database and Web Content Management Tools (9)

Installation: Apache & MySQL: Software Prerequisites, Installing Apache and PHP, Starting and Testing Apache, Testing PHP with phpinfo(), Installing MySQL, Starting and Testing MySQL, 125-145 Installing the php-mysql Module, Checking the php-mysql Module.

Using MySql: designing & creating your web database, working with mysql database, accessing mysql database from web with PHP, Advanced mysql administration, advanced mysql programming, Build your own PHP & MySql project website.

Web Content management System: Introduction, Wordpress, Drupal, Joomla.

Text Books

1. PHP 6 and MySQL 6 Bible by Steve Suehring, Wiley Publication.
2. Chris Bates, Web Programming Building Internet Applications, Second Edition, Wiley (2007)

References

1. Beginning PHP and MySQL by W. Jason Gilmore by Apress.
2. Complete Reference PHP by Steven Holzner, TMH.
3. Head First PHP and MySQL by Lynn Beighley, Michael Morrison, O'Reilly.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination
Examination Scheme:

Components	Internal	Mid Term	ESE	Total
Weightage (%)	30%	20%	50%	100%

Relationship between the Course Outcomes (COs), Program Outcomes (POs) and Program Specific Objectives(PSOs)

Course Outcomes	PO1	PO2	PO3	PO 4	PO 5	PO6	PO 7	PO8	PO9	PO 10	PSO1	PSO2
CO1												
CO2												
CO3												
CO4												
CO5												
Average												

1=weak 2= moderate 3=strong

	Software Engineering and Project Management	L	T	P	C
Version 1.0		3	0	0	3
Pre-requisites/Exposure	Basic knowledge of software development				
Co-requisites	--				

Course Objectives

1. To introduce the fundamental concepts of software engineering.
2. To build an understanding on various phases of software development.
3. To introduce various software process models.

Course Outcomes

On completion of this course, the students will be able to

CO1. Use knowledge of Software Life Cycle to successfully implement the projects in the Corporate world.
CO2. Identify the Inputs, Tools and techniques to get the required Project deliverable and Product deliverable
CO3. Implement Project Management Processes to successfully complete project in IT industry
CO4. Discuss Project management activities.

Catalog Description

Software engineering project management focuses on techniques for managing software engineering projects, and builds on core software engineering concepts. This course provides an understanding of both theoretical and methodological issues involved in modern software engineering project management and focuses strongly on practical techniques.

Course Content

Unit I: Introduction to Software Engineering and Project Management

Introduction to Software Engineering: Software, evolving role of software, three "R"-Reuse, Reengineering and Retooling, An Overview of IT Project Management: Define project, project management framework, the role of project Manager, Systems View of Project Management, Stakeholder management, Project phases and the project life cycle.

Unit II: Software Development Life Cycle Models

Overview of Software Development Life Cycle, Process Models: Waterfall Model, Evolutionary Process Model: Prototype and Spiral Model, Incremental Process model: Iterative approach, RAD, JAD model, Concurrent Development Model, Agile Development: Extreme programming, Scrum.

Unit III: Software Requirement Analysis and Specification

Types of Requirement, Feasibility Study, Requirement Analysis and Design: DFD, Data Dictionary, Requirement Elicitation: Interviews, Questionnaire, Brainstorming, Facilitated Application Specification Technique (FAST), Use Case Approach. SRS Case study, Software Estimation: Size Estimation: Function Point (Numerical). Cost Estimation: COCOMO (Numerical), COCOMO-II (Numerical), Earned Value Management.

Unit IV: Software Project Planning

Business Case, Project selection and Approval, Project charter, Project Scope management: Scope definition and Project Scope management, Creating the Work Breakdown Structures, Scope Verification, Scope Control.

Unit V: Project Scheduling and Human Resource management

Relationship between people and Effort: Staffing Level Estimation, Effect of schedule Change on Cost, Degree of Rigor & Task set selector, Project Schedule, Schedule Control, CPM (Numerical), Human Resource Planning, Acquiring the Project Team, Resource Assignment, Loading, Leveling, Developing the Project Team: Team Structures, Managing the Project Team.

Unit VI: Software Quality and Risk Management

Software quality, software reliability models, Overview of ISO 9001, SEI Capability Maturity Model, McCalls Quality Model, Six Sigma, Formal Technical Reviews, Tools and Techniques for Quality Control, Quality Control Charts, Modern Quality Management, Risk Management: Identify IT Project Risk, Risk Analysis and Assessment, Risk Strategies, Risk Monitoring and Control, Risk Response and Evaluation.

Unit VII: Software Maintenance

Maintenance Process, Maintenance Model, Estimation of maintenance cost, Regression Testing, Reverse Engineering, Software Re-engineering, Configuration Management and Documentation.

Text Book

- Software Engineering, 5th and 7th edition, by Roger S Pressman, McGraw Hill publication
- Software Engineering Project Management by Richard H. Thayer Wiley India Publication.

References:

- https://www.rgpvnotes.in/2018/01/cs-6003-software-engineering-project.html

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination
Examination Scheme:

Components	MSE	Quiz/Assignment/ etc.	ESE

Weightage (%)	20%	30%	50%

Relationship between the Course Outcomes (COs), Program Outcomes (POs) and Program Specific Outcomes (PSOs):

Course Outcomes	PO 1	PO 2	PO3	PO4	PO5	PO6	PO7	PO8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2
CO1														
CO2														
CO3														
CO4														
Average														

1.Weak 2. Moderate 3. Strong

	Business Communication and Ethics	L	T	P	C
Version 1.0		3	0	0	3
Pre-requisites/Exposure	--				
Co-requisites	--				

Course Objectives

1. Students will demonstrate conceptual domain knowledge of the sector.
2. Students will apply communication-support tools to decision making in sector.
3. Students will apply conceptual knowledge of communication in sector in an integrated manner.
4. Students will demonstrate employable and deployable skills for appropriate roles in communication

Course Outcomes

On completion of this course, the students will be able to

CO1. Students will demonstrate a clear understanding of the legal concepts related to communication and its importance in the concerned sector.
CO2. Students will be able to take appropriate decisions related to communication by apply strategic management knowledge.
CO3. Students will be able to analyses evolution of communication framework in media and is role in democratization process.
CO4. Students will integrate the conceptual connection between communication management and Ethics.

Catalog Description

Course Content

Unit 1: Communication skills in Business Management
Importance of communication skills in Business Management. Types of communication: the media and tools of communication. The Communication Process. Barriers and Gateways to communication. Ethics. Ethical models and Theories. Professional and Personal Ethics**,** Ethical Decision Making Matrix and Model, Personal language and body language.

Unit II: Types of communication
Types of managerial speeches: occasional speech; thematic speech. Para language; group discussions, meetings, seminars, and conferences. Art of facing interviews in: selection or placement, appraisal, Group Communication in: disciplinary committees and exit interviews. Formats for business letters and memos: routine type; sales promotion, bill collection. Formats for business letters and memos: disciplinary action; persuasive messages; negative messages.

Unit III: Ethics

Job application, preparing a professional resume and cover letter, follow-up messages and letters. Internal communication through: memos, minutes, notices, circulars. Writing effective Business Reports; Digital Communication.

Unit IV: Types of Ethics, Theories, models

PowerPoint preparation, Recruitment and Employment Correspondence Drafting the Employment Notice, Job Application Letter; Curriculum Vitae/ Resumes; Joining Interview; An offer of employment; Job Description; Letter of Acceptance, Letter of Resignation and Promotion, Testimonials and References, Business and Social Etiquette Professional conduct in a business setting: workplace hierarchy; the proper way to make introductions;

Unit V: Verbal and Non-verbal Communication

Use of courteous phrases and language in the workplace. Professional Image: appropriate business attire, Telephone Etiquette; Table etiquette

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination
Examination Scheme:

Components	Internal	Mid Term	ESE	Total
Weightage (%)	30%	20%	50%	100%

Relationship between the Program Outcomes (POs), Program Specific Outcomes and Course Outcomes (COs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PSO1	PSO2
CO1												
CO2												
CO3												
CO4												
Average												

1. WEAK 2. MODERATE 3. STRONG

	Problem Domains of AI	L	T	P	C
		3	0	0	3
Pre-requisites/Exposure	**--**				
Co-requisites	**--**				

Course Objectives

5. To provide the most fundamental knowledge of what the AI.
6. To learning how to solve problems through AI

.

Course Outcomes

On completion of this course, the students will be able to

CO1. Understand the basics of the theory and practice of Artificial Intelligence as a discipline and capable of problem formulation.

CO2. Formulate and solve given problem using Logic Programming and Knowledge representation

CO3. Design and analyze different issues in Expert System and Applications

CO4. Apply machine learning, natural language processing and neural network learning for solving AI problems

Catalog Description

Artificial intelligence (AI) is a research field that studies how to realize the intelligent human behaviors on a computer. The main topics in AI include problem solving, reasoning, planning, natural language understanding, Expert system, ANN, machine learning, NLP and so on.

Course Content

Unit 1: Introduction

Philosophy of artificial intelligence, Definitions - Evolution of AI - Applications of AI, Classification of AI- Intelligent Agents: Agents and Environment-Nature of Environment Structure Environment , Philosophical and ethical issues.

Unit II: Logic Programming and Knowledge Representation

Logic Programming**:** Introduction, Propositional Calculus Propositional, Logic, Natural Deduction System, A System in Propositional Logic, Resolution, Refutation in Propositional Logic, Predicate Logic, Logic Programming. **Knowledge Representation:** Introduction, Approaches to knowledge, Representation, Knowledge Representation using Semantic Network,

Unit III: Expert System and Applications: Introduction, Phases in Building Expert Systems Expert System Architecture, Expert Systems Vs Traditional Systems, Truth Maintenance Systems, Application of Expert Systems, List of Shells and tools.

Unit IV : Machine - Learning Paradigms: Introduction, Machine learning System, Supervised

and Unsupervised Learning, Inductive Learning, Learning Decision Trees, Deductive Learning, Clustering, Support Vector Machines.

Unit V: Artificial Neural Networks and Natural Language Processing: Introduction Artificial Neural Networks, Single - Layer Feed Forward Networks, Multi - Layer Feed Forward Networks, Radial - Basis Function Networks, Design Issues of Artificial Neural Networks, Recurrent Networks.

Advanced Knowledge Representation Techniques: Case Grammars, Semantic Web. Natural Language Processing: Introduction, Sentence Analysis Phases, Grammars and Parsers, Types of Parsers, Semantic Analysis, Universal Networking Knowledge.

Suggested Reading:

1. Saroj Kaushik, Artificial Intelligence, Cengage Learning India, First Edition, 2011.
2. Rich, Knight, Nair, Artificial Intelligence, Tata McGraw Hill, 3rd Edition 2009.

Textbook

1. Russel,S., and Norvig,P., (2015), Artificial Intelligence: A Modern Approach, 3rd Edition, Prentice Hall

Self-Learning Material

1. Department of Computer Science, University of California, Berkeley, http://www.youtube.com/playlist?list=PLD52D2B739E4D1C5F
2. NPTEL: Artificial Intelligence, https://nptel.ac.in/courses/106105077
3. Introduction to Artificial Intelligence (AI) (https://www.coursera.org/learn/introduction-to-ai)

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination
Examination Scheme:

Components	Internal	Mid Term	ESE	Total
Weightage (%)	30%	20%	50%	100%

Relationship between the Program Outcomes (POs), Program Specific Outcomes and Course Outcomes (COs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PSO1	PSO2
CO1												
CO2												

CO3												
CO4												
Average												

1. WEAK 2. MODERATE 3. STRONG

SLLS 7001	Programming in Data Structures	L	T	P	C
Version 1.0		4	0	0	4
Pre-requisites/Exposure	C language programming				
Co-requisites	--				

Course Objectives

1. To make students aware about the concept of data structures.
2. Implement linear and non-linear data structures by using C programming
3. To enable students to implement the concept of data structure using C programming language.

Course Outcomes

On completion of this course, the students will be able to

CO1. Design algorithms to perform operations with Linear and Nonlinear data structures
CO2. Illustrate various searching, Sorting and hashing techniques with their complexities analysis.
CO3. Exhibit a clear understanding of Hash Tables, Binary trees and Binary Search trees.
CO4. Choose appropriate data structures to solve real-world problems efficiently

Catalog Description

Knowledge about programming in C language is the building block of the students to build their programming skills. Students will be awarded based on the theoretical concepts of data structures and implementations in C programming. Topics to be covered include array, pointers, linked list, stack, queue, tree (traversals & operations), binary tree, binary search tree, AVL tree, M-way tree, searching and sorting techniques, hashing, graphs and graph traversals.

Course Content

Unit1: Introduction **5 lectures**

Basic Terminology, linear and non-linear data structures, elementary data organization, Structure operations, Algorithm Complexity and Time-Space trade-off, O-notation, Omega notation, and theta notation.

Unit 2: Linked List **7 Lectures**

Representation and Implementation of Singly Linked Lists, Two-way Header List, Traversing and Searching of Linked List, Overflow and Underflow, Insertion and deletion to/from Linked Lists, Insertion and deletion Algorithms, Doubly linked list, Linked List in Array, Polynomial representation, Generalized linked list, Garbage Collection and Compaction.

Unit 3: Stacks & Queues **8 lectures**

Array Representation and Implementation of stack, Operations on Stacks: Push & Pop,Linked Representation of Stack, Operations Associated with Stacks, Application of stack: Conversion of Infix to Prefix and Postfix Expressions, Evaluation of postfix expression using stack.

Array and linked representation and implementation of queues, Operations on Queue: Create, Add, Delete, Full and Empty, Circular queues, D-queues and Priority Queues.

Unit 4: Trees **8 Lectures**

Basic terminology, Binary Trees, Binary tree representation, algebraic Expressions, Complete Binary Tree, Extended Binary Trees, Array and Linked Representation of Binary trees, Traversing Binary trees, Threaded Binary trees, Traversing Threaded Binary trees, Huffman algorithm, AVL Trees.

Unit 5 : Searching and Sorting **10 Lectures**

Sequential search, binary search, comparison and analysis, Hash Table, Hash Functions, Collision Resolution Strategies, Hash Table Implementation, Linear probing, Quadratic, Double hashing, Bucket hashing, Deletion, and rehashing. Binary Search Tree (BST), Insertion and Deletion in BST, Path Length, B-trees. Insertion Sort, Bubble Sorting, QuickSort, radix sort Two-way Merge Sort, Heap Sort, Sorting on Different Keys, Practical consideration for Internal Sorting.

Unit 6: Heaps and Graphs **10 Lectures**

Heaps: Structure, Basic algorithms – Reheap Up, Reheap Down, Build heap, Insert, Delete Terminology & Representations, Graphs & Multi-graphs, Directed Graphs, Sequential -representations of Graphs, Adjacency Matrices, Traversal, Connected Component and Spanning Trees, Minimum Cost Spanning Trees.

Text Books

1. Data Structures with C (Schaum's Outline Series)by Seymour Lipschutz.
2. Let us C Solutions by Yashavant P. Kanetkar

Reference Books

1. Programming in ANSI C Book by E. Balagurusamy
2. Ellis Horowitz and Sartaz Sahni, “Data Structure using C”.

Modes of Evaluation: Quiz/Assignment/ Presentation/ Extempore/ Written Examination

Examination Scheme:

Components	Internal	Mid Term	ESE	Total
Weightage (%)	30%	20%	50%	100%

Relationship between the Course Outcomes (COs), Program Outcomes (POs) and Program Specific Objectives (PSOs)

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3

CO1	1		2	1								1	1		
CO2	1	1	2	1									2		
CO3	1	1	2	1								2	3		
CO4	1	1	1										1		
CO5	1	2	3	2								2	3		
Average	1	1.25	2	1.25								1.67	2		

1= Weak 2= Moderate 3= Strong

	Python programming Lab	L	T	P	C
Version 1.0		**0**	**0**	**2**	**2**
Pre-requisites/Exposure	--				
Co-requisites	--				

Course Objectives

1. To understand the fundamental of python programming language.
2. To write python scripting elements such as variables and flow control structures.
3. To use python library packages for solving domain problems.

Course Outcomes

CO1. Know the basic python programming concepts, data structures & regular expressions.

CO2. Discuss file-handling operations and understand OOPS concepts using python.

CO3. Develop modules and implement web development framework.

CO4. Discuss the role of python in advance technology.

Catalog Description

This course introduces the basic concepts of procedural and object-oriented programming using python programming language. This course also provides practical knowledge and hands-on experience in designing and implementing data structures. Activities covered include introduction to python programming language, datatypes, operators, loop structures, decision-making statements, fundamental data structures, functions, Classes and Objects, Constructor, File Handling, Exception Handling and Numpy module.

List of Experiments

Lab Exercise	Contents
Experiment No 1	Introduction
Experiment No 2 & 3	Basic Data Types
Experiment No 4	Strings
Experiment No 5 & 6	Functions
Experiment No 7 & 8	Files

Experiment No 9	Exceptions
Experiment No 10 & 11	OOPS
Experiment No 12 & 13	GUI using tkinter library
Experiment No 14	Usage of numpy
Experiment No 15	Usage of pandas

Experiment 1 - Introduction

Q1. Given an integer *n* perform the following conditional actions:

- If *n* is odd, print Weird
- If *n* is even and in the inclusive range of 2 to 5 , print Not Weird
- If *n* is even and in the inclusive range of 6 to 20, print Weird
- If *n* is even and greater than 20, print Not Weird

Q2. WAP to read an integer 'n' from STDIN. For all non-negative integers i<n, print i^2 on a separate line.

Sample input

3

Sample Output

1
2
4

Example

The list of non-negative integers that are less than 3 is [0, 1, 2]. The squares of each number is given below:

0
1
4

Q3. WAP to read an integer from STDIN. Without using any string methods, print the following on a single line:
123…n

Note that … represents the consecutive values in between.

Example

n=5
Output- 12345

Experiment 2 & 3– Basic Data Types

Q1. WAP to read the record of *n* students in a dictionary containing key/value pairs of name: [marks]. Print the average of the marks obtained by the particular student correct to 2 decimal places.

Input Format
The first line contains the integer *n*, the number of student's records. The next *n* lines contain the names and marks obtained by a student, each value separated by a space.

Sample Input
3
Krishna 67 68 69
Arjun 70 98 63
Malika 52 56 60

Sample Output
56.00

Q2. WAP to input a list of scores for N students in a list data type. Find the score of the runner-up and print the output.

Sample Input
N = 5
Scores= 2 3 6 6 5

Sample output
5

Note:
Given list is [2, 3, 6, 6, 5]. The maximum score is 6, second maximum is 5. Hence, we print 5 as the runner-up score.

Q3. Rupal has a huge collection of country stamps. She decided to count the total number of distinct country stamps in her collection. She asked for your help. You pick the stamps one by one from a stack of country stamps. Find the total number of distinct country stamps using a suitable data type.

Note: Apply your knowledge of the *.add()* operation from set to help your friend Rupal.

Experiment 4- Strings

Q1. WAP to enter a string and a substring. You have to print the number of times that the substring occurs in the given string. String traversal will take place from left to right, not from right to left.

Sample Input
ABCDCDC
CDC

Sample Output

2

Q2. WAP to input the first name, middle and last name of a person. Your task is to print the initials of the first and middle name separated by a dot (.)
The last name should be followed by a dot and a space where the first letter is capital.

Sample Input
Mohandas KaramChand Gandhi

Sample Output
M.K. Gandhi

Q3. Given a string containing both upper and lower case alphabets. Write a Python program to count the number of occurrences of each alphabet (case insensitive) and display the same.

Sample Input
ABaBCbGc

Sample Output
2A
3B
2C
1G

Experiment 5 & 6- Functions

Q1. Using functions, re-write and execute Python program to:

1. Add natural numbers upto n where n is taken as an input from user.
2. Print Fibonacci series till nth term (Take input from user).

Q2. At an airport, a traveler is allowed entry into the flight only if he clears the following checks:

1. Baggage Check
2. Immigration Check
3. Security Check

The logic for the check methods are given below:
check_baggage (baggage_weight)

- returns True if baggage_weight is greater than or equal to 0 and less than or equal to 40. Otherwise returns False.

check_immigration (expiry_year)

- returns True if expiry_year is greater than or equal to 2030 and less than or equal to 2050. Otherwise returns False.

check_security(noc_status)

- returns True if noc_status is 'valid' or 'VALID', for all other values return False.

traveler()

- Initialize the traveler Id and traveler name and invoke the functions check_baggage(), check_immigration() and check_security() by passing required arguments.
  Refer the table below for values of arguments.

Variable	Value
traveler_id	1001
traveler_name	Jim
baggage_weight	35
expiry_year	2019
noc_status	VALID

- If all values of check_baggage(), check_immigration() and check_security() are true,
  - display traveler_id and traveler_name
  - display "Allow Traveler to fly!"

  Otherwise,
  - display traveler_id and traveler_name
  - display "Detain Traveler for Re-checking!

Invoke the traveler() function. Modify the values of different variables in traveler() function and observe the output.

Q3. Write a Python program to find the maximum and minimum values in a given list of tuples using lambda function.

Original list with tuples:
[('V', 62), ('VI', 68), ('VII', 72), ('VIII', 70), ('IX', 74), ('X', 65)]
Output-
Maximum and minimum values of the said list of tuples:
(74, 62)

Experiment 7 & 8 - Files

Q1. Write a Python program to:

1. read a file.
2. add backslash (\) before every double quote in the file contents.
3. write it to another file in the same folder.
4. print the contents of both the files.

For example:

If the first file is 'TestFile1.txt' with text as:
Jack said, "Hello Pune".
The output of the file 'TestFile2.txt' should be:
Jack said,\"Hello Pune\".

Q2. Consider a file 'rhyme.txt' in D Drive with following text:

```
Jingle bells jingle bells
Jingle all the way
Oh what fun it is to ride
In a one horse open sleigh
Jingle bells jingle bells
Jingle all the way
```

Write a Python program to count the words in the file using a dictionary (use space as a delimiter). Find unique words and the count of their occurrences (ignoring case). Write the output in another file "words.txt" at the same location.

Q3. Assume a file city.txt with details of 5 cities in given format (cityname population(in lakhs) area(in sq KM)):
Example:
Dehradun 5.78 308.20
Delhi 190 1484
……
Open file city.txt and read to:

a. Display details of all cities
b. Display city names with population more than 10Lakhs
c. Display sum of areas of all cities

Experiment 9 -Exceptions

Q1. Input two values from user where the first line contains N, the number of test cases. The next N lines contain the space separated values of a and b. Perform integer division and print a/b. Print the error code in the case of *ZeroDivisionError* or *ValueError*.

Sample input
1 0
2 $
3 1

Sample Output
Error Code: integer division or modulo by zero
Error Code: invalid literal for int() with base 10: '$'
3

Q2. Assume the following Python code-
Rewrite the code to handle the exceptions raised. Print appropriate error messages wherever applicable.

```
mylist = [1,2,3,"4",5]
sum = 0
for i in mylist:
    sum = sum + i
print(sum)
print(mylist[5])
```

Q3. You have already created a Python program to implement the following in file handling section:

1. read a file.
2. add backslash (\) before every double quote in the file contents.
3. write it to another file in the same folder.
4. print the contents of both the files.

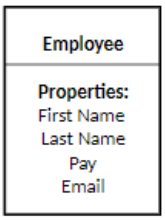
For example:
If the first file is 'TestFile1.txt' with text as:
Jack said, "Hello Pune".
The output of the file 'TestFile2.txt' should be:
Jack said,\"Hello Pune\".

Modify your code to implement Exception handling. Print appropriate error messages wherever applicable.

Experiment 10 & 11- OOPS

Q1.

- Create a class Employee with following properties
  - First Name
  - Last Name
  - Pay
  - Email : should be automatically generated as
    - Firstname + '.' + Lastname + "@company.com"
- Test the code with following information of an Employee:
  - First name is : ***Mohandas***
  - Last name is : ***Gandhi***
  - *Pay is* : ***50000***



Q2. Perform the following instructions:

a) Create a Vehicle class with max_speed and mileage as instance attributes. Additionally, create a method named seating_capacity() using the below syntax:

```
def seating_capacity(self, capacity):
    return f"The seating capacity of a {self.name} is {capacity} passengers"
```

b) Create child class 'Bus' that will inherit all of the variables and methods of the Vehicle class. Set the seating capacity of the bus to 50 using super().
c) Create a Bus object that will inherit all of the variables and methods of the Vehicle class and display it.
d) Define a class attribute "color" with a default value white. I.e., Every Vehicle should be white.

Q3.

Q: List the risk associated with the implementation of Account class. Suggest a solution.

```
class Account:
    def __init__(self, initial_amount):
        self.balance = initial_amount
    def withdraw(self,amount):
        self.balance = self.balance - amount
    def deposit(self,amount):
        self.balance = self.balance + amount
ac = Account(1000)
ac.balance = 2000 #stmt1
ac.balance = -1000 #stmt2
print(ac.balance) #stmt3
```

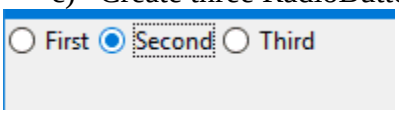
Experiment 12 & 13- GUI using tkinter library

Q1.

a) import Tkinter package and create a window and set its title
b) set the default window size using geometry function
c) Create a label with "Hello" text in it and set its position on the form.
d) Add a button to the window with "CLICK ME" written on it.
e) change the foreground and background color for the button created above
f) Create a function that will be executed when the button is clicked and print "Button was clicked" on clicking the button

Q2. This is the continuation of Question1, add the given below features in the above program:

a) Take **user name** as input using the Tkinter Entry class
b) Print the entered text (username) on clicking the button.
c) Create three RadioButtons as displayed below



d) Print the currently selected radio button or the radio button value.

Q3. Write a program to accept following details from a student using GUI
1. Name of the student (using Textbox)
2. Gender (Using radio button)
3. Qualification (Using List)
4. Marks of three subjects (using Textbox)
Compute the percentage of the student and display it in a textbox.

Experiment 14 - Numpy

Q1.

a. Convert numbers =[1, 2.0, 3] to numpy array and convert all elements to string type.
b. Create a 2 D array through list and set dtype as int32
c. Find the rows and columns of the 2d array created in part b
d. Print 10 random numbers between 1 and 100.

Q2.

a) Write a NumPy program to get help on the add function
b) Write a NumPy program to test whether none of the elements of a given array is zero
c) Write a NumPy program to test whether any of the elements of a given array is non-zero
d) Write a NumPy program to generate an array of 15 random numbers from a standard normal distribution

Experiment 15 - pandas

Q1. Refer the given excel file and perform various operations using pandas library:

0	GOOGL	27.82	87	845	larry page
1	WMT	4.61	484	65	n.a.
2	MSFT	-1	85	64	bill gates
3	RIL	not available	50	1023	mukesh ambani
4	TATA	5.6	-1	n.a.	ratan tata

a. Read the above excel file in python.

b. How do I write this file to a new file "new.csv"?

c. Include column names in this file. Use 'ticker', 'eps', 'revenue', 'price', 'people' as column names.

d. Convert all not available or n.a. values to NAN and also convert negative revenues to NAN because revenues can never be negative.

e. Fill NAN values using a suitable approach.

f. Write a function to change n.a value appearing in WMT to Sam Walton

Text Books

1. Python Programming by IBM

Reference Books

2. Python Programming by "Reema Thareja"
3. Learning Python – Mark Lutz, O'reilly publication

Continuous Evaluation: There will be continuous evaluation for all practical subjects of SCS during the semester. The performance of a student in a Practical subject will be evaluated as per process given below:

Components of evaluation

a. Viva voce / Quiz (50%) + Performance & Records (50%).
b. Lab performance and record evaluation shall be a continuous process throughout the semester.
c. Minimum three Viva voce/ Quiz based on practical sessions shall be conducted during the semester.

Relationship between the Course Outcomes (COs), Program Outcomes (POs) and Program Specific Outcomes (PSOs):

Course Outcomes	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO1	PSO2	PSO3
CO1															
CO2															
CO3															
CO4															

1 = Weak 2 = Moderate 3 = Strong

	Web Technology through PHP Lab	L	T	P	C
Version 1.0		0	0	1	1
Pre-requisites/Exposure	Basic knowledge of web programming				
Co-requisites	--				

Course Objectives

The Course objective is to provide overview on Web technologies and Server Side programming. Upon the completion of this course, the students will be having the knowledge on Web technologies, and will be able to develop the Dynamic Web Programs using PHP and databases.

Course Outcomes

On completion of this course, the students will be able to

CO1. Learn markup languages HTML and scripting languages JavaScript
CO2. Identify the basic PHP programming structures
CO3. Develop real world object-oriented concepts using PHP
CO4. Develop interactive web based program with database connectivity

Catalog Description

Nowadays, the use of web applications are enormous by the individuals and organizations. Companies are using web-based solutions for their customers. Companies are also using CRM software to increase the loyalty of their consumers, and for taking feedbacks from their customers. Banks are using web based solutions to facilitate their customer to do transections, and access required information from anywhere in the world. The use of social and professional networking web sites are increasing day-by-day. For the development of web-based application, companies are recruiting smart-working employees with knowledge of server-side scripting language like PHP and web technologies. In this course, students will be able to understand the different web technologies. They would learn how to develop real world web-based applications using PHP programming. They will also learn how to develop the projects as a team.

List of Experiments

Experiment 01 - HTML: To design the web pages in HTML using Lists, Tables, Images, Forms and Frames.
Experiment 02 - JavaScript: To develop simple dynamic web pages to validate data using messages and confirmations.
Experiment 03 - Arrays and Functions in JavaScript: To develop simple web pages using input and output statements, functions and arrays.
Experiment 04 - Exception Handling: To develop simple web programs with exception handling.

Experiment 05 - PHP Basics: Installation of PHP and use of basic control structures.
Experiment 06 - String Manipulation in PHP with Form Validation: To develop web pages using PHP to perform String Manipulations and Form Validations.
Experiment 07 - Classes and Objects in PHP: To develop simple web pages to understand the concepts of classes and objects.
Experiment 08 - File Operations in PHP: To develop programs to perform the various file operations in PHP.
Experiment 09 - Exception Handling in PHP: To develop simple programs to understand the concept exceptional handling in PHP.
Experiment 10 - Database Connectivity: To develop simple programs to connect the webpages with the database.
Experiment 11 - Content Management Tools: To work with any one of the content management tool.

Continuous Evaluation- There will be continuous evaluation for all practical subjects of SCS during the semester. The performance of a student in a Practical subject will be evaluated as per process given below:

Components of evaluation

a. Viva voce / Quiz (50%) + Performance & Records (50%).
b. Lab performance and record evaluation shall be a continuous process throughout the semester.
c. Minimum three Viva voce/ Quiz based on practical sessions shall be conducted during the semester.

Text Books

1. PHP 6 and MySQL 6 Bible by Steve Suehring, Wiley Publication.
2. Chris Bates, Web Programming Building Internet Applications, Second Edition, Wiley (2007)

References

1. Beginning PHP and MySQL by W. Jason Gilmore by Apress.
2. Complete Reference PHP by Steven Holzner, TMH.
3. Head First PHP and MySQL by Lynn Beighley, Michael Morrison, O'Reilly.

	Data Structures Lab	L	T	P	C
Version 1.0		0	0	2	1
Pre-requisites/Exposure	Basic knowledge of C programming and Data Structures.				
Co-requisites	--				

Course Objectives

1. To enable students to develop problem solving skills.
2. To make students aware about the concept of linear and non-linear data structures.
3. To enable students to implement the sorting/searching algorithms and file handling.

Course Outcomes

Upon completion of this course the learners will be able to:

CO1. Implement elementary data structures like arrays, structures and linked list.

CO2. Implement Stack and queue with array and linked list.

CO3. Use searching and sorting algorithms and file handling.

CO4. Implement non-linear data structures to solve real world problems

Catalog Description

This course provides practical knowledge and hands-on experience in designing and implementing the elementary data structures. Activities covered include pointers, arrays, linked lists, primitive and non-primitive data structures, sorting, searching, hashing algorithms, trees and graphs.

List of Experiments

Experiment-1: Array & Structure

To apply the concept of array, structure and experiment on nested array and array of structures.

Experiment-2: Union and Dynamic Memory Allocation

To implement the concept of union and experiment on dynamic memory allocation.

Experiment-3: Link List Data Structure and its Applications

To experiment the concept of pointers, structure and dynamic memory allocation to realize linked list, its types and application.

Experiment-4: Stack Data Structure
To demonstrate use of arrays and linked list to implement Stack operations and applications of Stack.

Experiment-5: Queue Data Structures
To demonstrate use of arrays and linked list to implement Queue operations and types of Queues.

Experiment-6: Sorting algorithms
To implement various sorting algorithms.

Experiment-7: Searching algorithms
To implement linear search, binary search.

Experiment-8: Hash Tables
To implement hash tables with and without collision avoidance algorithms using arrays/linked lists.

Experiment-9: Trees
To demonstrate creation of a binary tree using arrays/linked lists and working with tree traversal and heap sorting algorithms.

Experiment-10: Advanced Trees data structures
To implement AVL Trees¸ heap sort, priority queue

Experiment-11: Graphs-I
To implement graph traversing techniques.

Experiment-12: Graphs-II
To show the representation of graphs using adjacency matrix.

Text Books

1. Seymour Lipschutz, “Data Structures with C (Schaum's Outline Series)”.
2. Yashavant P. Kanetkar, “Let us C”.

Reference Books

1. E. Balagurusamy, “Programming in ANSI C”.
2. Ellis Horowitz and SartazSahni, “Data Structure using C”.

Continuous Evaluation

There will be continuous evaluation for all practical subjects of SoCS during the semester. The performance of a student in a Practical subject will be evaluated as per component of evaluation given below:

1. Viva voce / Quiz (50%)
2. Performance & Records (50%).

Lab performance and record evaluation shall be a continuous process throughout the semester. Minimum two Viva-voce and two Quizzes based on practical sessions shall be conducted during the semester.

Relationship between Program Outcomes (POs), Program Specific Outcomes (PSOs) and Course Outcomes (COs)

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	1	1	2	1								2	3		
CO2	1	1	2	1								2	3		
CO3	1	1	2	1								2	3		
CO4	1	1	2	1								2	3		
Average	1	1	2	1								2	3		

1=weak 2= moderate 3=strong

SEMESTER II

	Database Management Systems	L	T	P	C
Version 1.0		3	0	0	3
Pre-requisites/Exposure					
Co-requisites	--				

Course Objectives

1. To understand the architecture and functioning of database management systems as well as associated tools and techniques, principles of data modelling using entity relationship and develop a good database design and normalization techniques to normalize a database.
2. To understand the use of structured query language and its syntax, transactions, database recovery and techniques for query optimization.
3. To acquire a good understanding of database systems concepts and to be in a position to use and design databases for different applications.
4. To be familiar with the contemporary database models like OO Databases, Distributed Databases etc.

Course Outcomes

On completion of this course, the students will be able to

CO1. Define the terminology, features, classifications, and characteristics of database systems.

CO2. Analyze an information storage problem and derive an information model expressed in the form of an entity relation diagram.

CO3. Demonstrate relational data model, transform an information model into a relational database schema.

CO4. Formulate using relational algebra, relational calculus and SQL solutions to a broad range of query problems and optimize query using various algorithms.

CO5. Demonstrate an understanding of normalization theory and apply such knowledge to the normalization of a database.

Catalog description:

A Database management system allows users to define, store, retrieve and update the information contained in the database on demand. Information can be anything that is of significance to an individual or organization. In this course, the focus will be on relational model, query writing and procession, Integrity constraints ER, EER Model, storage structures of database systems, transactional database systems & the techniques of concurrency control and recovery, contemporary databases. Students are encouraged to play an active role in the construction of their own knowledge and in the design of their own learning strategies.

Course Content

Unit I: Database concept **7 lecture hours**

Introduction , Data, Information Metadata, Terminology Of File, Association Between Fields, Entities And Their Attributes, Relationship Record And Files, Abstraction And Data Integration, Association Between Files(Record Types), Conventional File Processing System, Database System, Components Of Database Management System – (Classification Of DBMS Users, The Three-Level Architecture Proposal For DBMS, Mapping Between View, Data Independence.

Unit II: Data Models **8 lecture hours**

Introduction , Data Association-(Entities, Attributes And Associations, Relationship Among Entities, Representation Of Association And Relationship), Data Model Classification-(Approaches To The Relational Model, Hierarchical Model & Network Model With Examples), Entity – Relationship Model. Concept of File Organization – Sequential Files, Index-Sequential Files, Direct Files, Secondary Key Retrieval.

Unit III: Relational Model **7 lecture hours**

Relational Database: Attributes and Domains, Tuples, Relation and Their Schemas Relation Representation, Keys, Relationship, Relational Operations, Integrity Rules. Relational Algebra: Basic Operations, Relational Algebra Queries, And Relational Calculus: Tuple Calculus, Domain Calculus. Relational Database Manipulations: Introduction, SQL, Data Manipulations in SQL, QUEL and QBE.

Unit IV: Relational Database Design **8 lecture hours**

Relational Schema, Relational Design, Functional Dependency, Normalization, First –Second-Third Normal Forms, Relational With More Than One Candidates Key ,Good And Bad Decomposition, Multivalued Dependency, Fourth Normal Form, Fifth Normal Form.

Network Data Model: The Architecture of DBTG System, Schema & Subschema, and DBTG. Data Manipulation Facility. Hierarchical Data Model: The Tree Concept, Architecture of ANIMS System, Data Manipulation.

Unit V: Database Operations and Maintenance **6-lecture hour**

Database Administrator (DBA, Database Security, Integrity and Control (User with Password and Complete / Limited Authorization, Encryption of Data). Concurrency Control: Problem of Concurrent Access, Resource Looking, Deadlock. Database Recovery: Restore, Backward & Forward Recovery. Distributed Database: Introduction, Data Distribution, and Deadlock In Distributed Systems, Security, Security And Protection, Homogeneous And Heterogeneous Systems. Knowledge Base and Database Systems, Expert Database Systems, Object Database System.

Text Books

1. Elmasri, Navathe, "Fundamentals of Database Systems", Addison Wesley, 6th Edition, 2011

Reference Books

1. Korth, Silberschatz, Sudarshan, "Database Concepts", McGraw Hill, 6th Edition, 2010

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme:

Components	MSE	Presentation/Assignment/ etc	ESE
Weightage (%)	20%	30%	50%

Relationship between the Course Outcomes (COs), Program Outcomes (POs) and Program Specific Objectives (PSOs)

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PSO1	PSO2
CO1												
CO2												
CO3												
CO4												

CO5												
Avg												

1= Weak 2= Moderate 3= Strong

UNIVERSITY OF PETROLEUM & ENERGY STUDIES

	Operating Systems	L	T	P	C
Version 1.0		**4**	**0**	**0**	**4**
Pre-requisites/Exposure	**--**				
Co-requisites	**--**				

Course Objectives

1. To understand fundamentals of operating systems.
2. To understand and evaluate Process Management, Memory Management, Virtual Memory, File System.
3. To understand and evaluate I/O and security aspects of operating systems.

Course Outcomes

On completion of this course, the students will be able to

CO1. Explain various types of operating systems and its structure
CO2. Apply various process management and CPU scheduling algorithms.
CO3. Discuss various memory management techniques.
CO4. Analyze various disk-scheduling algorithms and file systems.
CO5. Apply various deadlock prevention, avoidance, and detection and recovery techniques.

Catalog Description

This course covers the classical internal algorithms and structures of operating systems. It introduces the core concepts of operating systems, such as processes and threads, scheduling, synchronization, memory management, file systems, input and output device management and security.

Course Content

Unit I: INTRODUCTION TO OPERATING SYSTEM 3 lecture hours

Introduction to OS, Its need and operating system services, Operating system Classification – Single user, Multi user, Simple batch Processing, Multi programming , Multitasking , Parallel systems, Distributed system & Real time system (overview).

Unit II: Process Management 14 lecture hours

Process Concept, Inter process communication- Race conditions –Critical Sections –Mutual Exclusion –Busy waiting – Sleep and Wakeup – semaphores- Event counter – Monitors- Message passing, Threads, Process scheduling & CPU scheduling – Round robin scheduling – priority scheduling – multiple queues- shortest job first- guaranteed scheduling- two –level scheduling.

Unit III: Deadlock **4 lecture hours**

Deadlock - conditions for deadlock. Deadlock detection and recovery. Deadlock avoidance - resource trajectories - safe and unsafe states - bankers' algorithm. Deadlock prevention. Two phase locking – non-resource deadlocks – starvation, security mechanism and policy, Domain of protection, access matrix.

Unit IV: Memory Management **8 lecture hours**

Logical versus Physical Address space, Swapping –Multiprogramming with fixed and variable partitions- Memory management with bit maps, linked list, buddy system- allocation of swap space. Virtual memory- paging and segmentation, page tables, associative memory- inverted page tables. Allocation algorithm, Page replacement algorithm, thrashing.

Unit V: File system **6 lecture hours**

File systems and I/O files. Directories- file system implementation- security and protection mechanisms. Principles of I/O hardware – I/O devices- device controllers-DMA. Principle of I/ O software – Interrupt handles- device drivers – Disk Scheduling- Clock and terminals. I/O buffering –RAID –Disk cache, FCFS scheduling , SSTF scheduling, SCAN Scheduling, C-SCAN scheduling, Selecting disk scheduling algorithms,

Unit VI CASE STUDY: UNIX / LINUX operating system **1 lecture hour**

Text Books

1. SILBERSCHATZ, Galvin (2010), Operating System Concepts 8e, Wiley India.
2. William Stallings, "Operating systems", Pearson Education, Fifth edition.
3. D.M. Dhamdhere, "Operating Systems", 2nd Edition, Tata McGraw-Hill.

Reference Books

1. Garry Nutt, "Operating Systems – A Modern perspective", Third Edition, Pearson Education.
2. Andrew S. Tanenbaum, "Modern Operating Systems", Prentice Hall.
3. Bach, M.J., "Design of UNIX Operating System", Prentice Hall.
4. Charles Crowley, "Operating systems – A Design Oriented Approach", Tata Mc Grawhill, 1997.
5. Michel Palmer "Guide o Operating Systems", Vikas Thomson Learning Publishing, NewDelhi.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme:

Components	MSE	Presentation/Assignment/ etc	ESE
Weightage (%)	20%	30%	50%

Relationship between the Course Outcomes (COs), Program Outcomes (POs) and Program Specific Objectives (PSOs)

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PSO1	PSO2
CO1												
CO2												
CO3												
CO4												
CO5												
Avg												

1= Weak 2= Moderate 3= Strong

	Object Oriented Analysis and Design using UML	L	T	P	C
Version 1.0		3	0	0	3
Pre-requisites/Exposure					
Co-requisites	--				

Course Objectives

1. The student should be able to design different UML diagrams like Use case diagram, class diagram and sequence diagrams.
2. To select and apply in autonomy appropriate technologies and techniques for different types of testing all over the software process development in different domains and contexts.

Course Outcomes

On completion of this course, the students will be able to

CO1. State the importance of object oriented modeling with UML.
CO2. Identify the role of actors, classes and objects in a software design.
CO3. Demonstrate the models for activity relationship and interaction modeling.
CO4. Analyze events and design state diagram.
CO5. Evaluate component and deployment diagrams.

Catalog Description

This course provides an in depth knowledge about the different Object-Oriented Analysis and Design (OOAD) strategies to architect and build object oriented systems using Unified Modeling Language (UML). The students will also learn how to consider any subject matter and identify all its concepts and relationships to express and model them using UML Analysis, and then design them in preparation for the programming phase. Furthermore, course also includes advanced Analysis and Design strategies based upon the most important Object-Oriented Patterns.

Course Content

Unit 1: OOAD basics **4 lecture hours**

Overview of object oriented system development, Object basics, The unified process, SDLC, Modelling concepts, Modeling as a design technique, Analysis and modelling, UML diagrams, Use case, Class ,State ,Interaction Modelling, Rational Unified Process (RUP).

Unit 2: Basic and advanced structural modeling **8 lecture hours**

Requirement Modeling :Requirement Engineering, Requirement Modeling: Use Cases, Use case Diagrams, Examples of UCD, Basic structural Modeling: Classes, Relationships, Common Mechanisms Diagrams, Interfaces, Types and Roles, Packages, Class & object Diagrams : Terms ,concepts, Modeling techniques for class & object diagrams, Examples of class and object diagram, Identifying operations and Specifying operations using CRC card

Unit 3: Basic behavioral modeling 8 **lecture hours**

Activity Diagram terms and concepts, Activity Diagrams: Examples, Swim lanes Design Architecture, Interaction, Diagrams (SEQUENCE, COLLABORATION), Interaction Diagrams (SEQUENCE, COLLABORATION), Practice examples on sequence and collaboration.

Unit 4: Advanced behavioral modeling 8 **lecture hours**

Events and Signals, State Machines, Processes and Threads, Time and Space, State chart diagrams & Sequential Diagrams, State chart diagrams & Sequential Diagrams.

Unit 5: Architectural modeling **8 lecture hours**

Terms and concepts of component and deployment, Component Diagrams and Examples, Deployment Diagrams and Examples.

Text Books

1. Object-Oriented Modeling and Design with UML, Michael R Blaha & James R Rumbaugh, Pearson Education.

2. Object Oriented Analysis and Design, Atul Kahate, TMH

References

1. Object-Oriented Analysis and Design with Applications, 3/e, Grady Booch, Jim Conallen, Michael W. Engel, Kelli A. Houston, Robert A. Maksimchuk and Bobbi J. Young

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination
Examination Scheme:

Components	Internal	Mid Term	ESE	Total
Weightage (%)	30%	20%	50%	100%

Relationship between the Program Outcomes (POs), Program Specific Outcomes and Course Outcomes (COs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PSO1	PSO2
CO1												
CO2												
CO3												
CO4												

Average												

1. WEAK 2. MODERATE 3. STRONG

	Venture ideation and Enterprenureship	L	T	P	C
Version 1.0		2	0	0	2
Pre-requisites/Exposure					
Co-requisites	--				

*Awaited from UCI

	Soft Computing	L	T	P	C
Version 1.0		3	0	0	3
Pre-requisites/Exposure	--				
Co-requisites	--				

Course Objectives

The Course objective is to provide overview on Soft Computing concepts, technologies, and applications. Upon the completion of this course, the students will be having the knowledge on the underlying principle of soft computing with its usage in various application and different soft computing tools to solve real life problems

Course Outcomes

On completion of this course, the students will be able to
CO1: Classify the various soft computing frameworks.
CO2: Be familiar with the design of neural networks, fuzzy logic and fuzzy systems
CO3: Develop application on different soft computing techniques like Fuzzy, GA and Neural network
CO4: Implement Neuro-Fuzzy and Neuro-Fuzz-GA expert system
CO5: Learn mathematical background for optimized genetic programming

Catalog Description

Soft computing is an emerging approach to computing which parallel the remarkable ability of the human mind to reason and learn in an environment of uncertainty and imprecision. Soft computing is based on some biological inspired methodologies such as genetics, evolution, ant's behaviors, particles swarming, human nervous systems, etc. Now, soft computing is the only solution when we don't have any mathematical modelling of problem solving (i.e., algorithm), need a solution to a complex problem in real time, easy to adapt with changed scenario and can be implemented with parallel computing. It has enormous applications in many application areas such as medical diagnosis, computer vision, hand written character recondition, pattern recognition, machine intelligence, weather forecasting, network optimization, VLSI design, etc.

Course Content

Unit I: 8 lecture hours
Concept of Computing Systems, Difference between Soft and Hard computing, Characteristics of Soft Computing, Requirement of Soft computing, Major Areas of Soft Computing, Applications of Soft Computing.

Unit 2: 6 lecture hours
What is Neural Network, Learning rules and various activation functions, Single layer Perceptron's, Back Propagation networks, Architecture of Backpropagation(BP) Networks,

Backpropagation Learning, Variation of Standard Back propagation Neural Network, Introduction to Associative Memory, Adaptive Resonance theory and Self Organizing Map, Recent Applications.

Unit 3: 9 lecture hours

Artificial Neural Networks: Supervised Learning: Introduction and how brain works, Neuron as a simple computing element, The Single layer Perceptron, Backpropagation networks: architecture, multilayer perceptron, backpropagation learning-input layer, accelerated learning in multilayer perceptron, The Hopfield network, Bidirectional associative memories (BAM), RBF Neural Network.

Artificial Neural Networks: Unsupervised Learning: Hebbian Learning, Generalized Hebbian learning algorithm, Competitive learning, Self- Organizing Computational Maps: Kohonen Network.

Unit 4: 5 lecture hours

Fuzzy Logic Crisp & fuzzy sets fuzzy relations fuzzy conditional statements fuzzy rules fuzzy algorithm. Fuzzy logic controller.

Unit 5: 8 lecture hours

Genetic algorithms basic concepts, encoding, fitness function, reproduction-Roulette wheel, Boltzmann, tournament, rank, and steady state selections, Convergence of GA, Applications of GA case studies. Introduction to genetic programming- basic concepts.

Textbooks

1. R. Rajasekaran and G. A and Vijayalakshmi Pa, *Neural Networks, Fuzzy Logic, and Genetic Algorithms: Synthesis and Applications*, Prentice Hall of India
2. D. E. Goldberg, *Genetic Algorithms in Search, Optimisation, and Machine Learning*, Addison-Wesley

Reference Books

1. L. Fausett, *Fundamentals of Neural Networks*, Prentice Hall
2. T. Ross, *Fuzzy Logic with Engineering Applications*, Tata McGraw Hill

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination
Examination Scheme:

Components	MSE	Presentation/Assignment/Test/Quiz etc.	ESE
Weightage (%)	20	30	50

Relationship between the Course Outcomes (COs), Program Outcomes (POs) and Program Specific Outcomes (PSOs).

Course Outcomes	PO1	PO2	PO3	PO 4	PO 5	PO6	PO 7	PO8	PO9	PO 10	PSO 1	PSO 2
CO1												
CO2												
CO3												
CO4												
CO5												
Average												

1=weak 2= moderate 3=strong

	Knowledge Engineering and Expert Systems	L	T	P	C
Version 1.0		3	0	0	3
Pre-requisites/Exposure					
Co-requisites	--				

Course Objectives

The main objective of this course is to provide the students with an understanding of the principles of knowledge engineering and the design and development, planning, and management of an expert system.

Course Outcomes

On completion of this course, the students will be able to
CO1: Explain and describe the concepts central to the creation of knowledge bases and expert systems.
CO2: Understand the methods used to evaluate the performance of an expert system
CO3: Conduct an in-depth examination of an existing expert system with an emphasis on basic methods of creating a knowledge base.
CO4: Examine properties of existing systems in a case-study manner comparing differing approaches.

Catalog Description

This course includes techniques for the construction of expert systems, including computer inference and knowledge acquisition; knowledge representation schemes validation and measurement methods. The students will be able to examine properties of existing systems in a case-study manner, comparing differing approaches.

Course Content

Unit I: Introduction **8 lecture hours**

The history of knowledge-based expert system, Characteristics of current expert systems, Basic concepts for building expert systems, the architecture of expert systems, constructing an expert system, Tools for building expert systems, evaluating an expert system, Languages and tools for knowledge engineering.

Unit 2: Knowledge Engineering and basics of expert systems **8 lecture hours**

Knowledge representation**, t**he meaning of an expert system, problem domain and knowledge domain, the advantages of an expert system, general stages in the development of an expert system,

general characteristics of an expert system, history and uses of expert systems today, rule-based expert systems, procedural and nonprocedural paradigms, characteristics of artificial neural systems. -The study of logic, difference between formal logic and informal logic, meaning of knowledge, how knowledge can be represented, semantic nets, how to translate semantic nets into PROLOG, limitations of semantic nets, schemas, frames and their limitations, how to use logic and set symbols to represent knowledge, the meaning of propositional and first order predicate logic, quantifiers, imitations of propositional and predicate logic.

Unit 3 **8 Lecture hours**

Trees, lattices, and graphs, state and problem spaces, AND-OR trees and goals, methods of inference, rules of inference, limitations of propositional logic, logic systems, resolution rule of inference, resolution systems, and deduction, shallow and causal reasoning, applying resolution to first-order predicate logic, forward and backward chaining, additional methods of reference, Meta knowledge, the Markov decision process.

Unit 4 **8 lecture hours**

The meaning of uncertainty and theories devised to deal with it, types of errors attributed to uncertainty, errors associate, with induction, features of classical probability, experimental and subjective probabilities, compound and conditional probabilities, hypothetical reasoning and backward induction, temporal reasoning, Markov chains, odds of belief, sufficiency and necessity, role of uncertainty in inference chains, implications of combining evidence, role of inference nets in expert systems, how probabilities are propagated.

Unit 5 **4 lecture hours**

A Case Study in Knowledge Engineering.

Textbooks

1. Jcafe, inc.: Guide Author Electronic Multimedia Publishing Tool, Student Edition 5.0
2. Rudloff, Winfried K.: “Hypertext/Multimedia-Based Lecture Notes”, 1998-2009

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination
Examination Scheme:

Components	MSE	Presentation/Assignment/Test/Quiz etc.	ESE
Weightage (%)	20	30	50

	JAVA Programming	L	T	P	C
Version 1.0		3	0	0	3
Pre-requisites/Exposure	• Basic Knowledge of C/C++ Programming. • Basic Knowledge of Object Oriented Design.				
Co-requisites	Knowledge of Computer Programming				

Course Objectives

The objectives of this course are to:

1. Create Java programs that leverage the object-oriented features.
2. Design & implement multithreading and data structure.
3. Learn the concepts of JDBC and Servlets.

Course Outcomes

On completion of this course, the students will be able to

CO1. Understand complete architecture of Java Platform and able to relate with OOPS concepts
CO2. Analyze real world object-oriented concepts and incorporate into the Java programming Language.
CO3. Implement and execute programs which are based on the OOPS concept.
CO4. Differentiate Java Programming from other OOPS based programming languages.
CO5. Develop basic level applications based on core java and advance java skills.

Catalog Description

Java is a programming language and computing platform, first released by Sun Microsystems in 1995. Many of applications and websites will not work unless you have Java installed, and more are creating every day. Java is fast, secure, and reliable. From laptops to datacentres, game consoles to scientific supercomputers, cell phones to the Internet, Java is everywhere!

Course Content

Unit I: [5 lecture hours]

Introduction

Feature of Java, JVM, JRE, class path, classes, fields, access control, objects creation, initialization, methods, this, overloading methods, main method, native methods, class design, lexical elements, types and literals, variables, array variables, naming, operators, expressions, member access, precedence, associativity, statements & blocks, if-else, switch, while and do-while, for, labels, break, continue, return, goto.

Unit II: [6 lecture hours]

Inheritance, Interfaces And Packages

Extended class, constructors in extended classes, inheriting and redefining members, type compatibility, conversion, protected, final methods and classes, abstract methods and classes,

Object class, cloning objects, designing extended classes, single inheritance versus multiple inheritance. Interface, interface declarations, extending interfaces, working with interfaces, marker interfaces, when to use interfaces. Package naming, type imports, package access, package contents, package objects and specifications.

Unit III: **[6 lecture hours]**

Exception And String Handling
Creating exception types, throw, throws, try, catch, finally, custom exception, when to use exception, Wrapper classes, loading classes. String operations, String comparisons, utility methods, making related strings, String conversions, Strings and char arrays, String and byte arrays, String Buffer, String Builder.

Unit IV:

NESTED CLASSES AND THREADS **[6 lecture hours]**

Static nested types, inner classes, local inner classes, anonymous inner classes, inheriting nested types, nesting in interfaces, implementation of nested types. Creating threads, using runnable, synchronization, wait, notify, notifyall, waiting and notification, thread scheduling, deadlocks, ending thread execution, thread management, security, and threadgroup, threads and exceptions, debugging threads.

Unit V: **[6 lecture hours]**
Collections, Design Pattern And Jdbc
Collections, iteration, Collection interface, set, sortedset, list, map, sortedmap, wrapped collections and collections class, arrays, legacy collection, properties. Object composition principles singleton design pattern, DAO design pattern, factory design pattern. JDBC: types of drivers, characteristic, components, database connectivity, Statement, Prepared Statement, CallableStatement, Resultset.

Unit VI: **[7 lecture hours]**
Advanced Java
Servlets: Introduction, Benefits, Architecture, GET, POST methods, Servlet container, Servlet's Life Cycle, ServletConfig, ServletContext, Requests & Responses, GenericServlet, Thread-Safe Servlets, HttpServlet Class, HttpServletRequest, HttpServletResponse interface, Deployment Descriptor; Session Management: URL Rewriting, Hidden Fields, Cookies, Session Objects, Servlet Filter, Servlet Listeners. JSP: Introduction, problem with servlets, how JSP work, implicit object, directives, scripting elements, comments, JSP life cycle Attributes: Application, request, session, page; web application deployment,
Security.

Text Books

T1: The Java Programming Language 3rd Edition, Ken Arnold, James Gosling, Pearson.
T2: Head First Servlets and JSP 2nd Edition.
T3: The Complete Reference Java 7th Edition, Herbert-Schild, TMH.

T4: Java SE7 Programmer I &II Study Guide, Kathy Sierra and Bert Bates, McGraw Hill.

	Database Management Systems Lab	L	T	P	C
Version 1.0		0	0	1	1
Pre-requisites/Exposure	--				
Co-requisites	--				

Reference Books

R1: A premier guide to SCJP 3rd Edition, Khalid Mughal, Pearson.
R2: Thinking in Java, 3rd Edition, Bruce Ackel, Pearson.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination
Examination Scheme:

Components	**MSE**	**Presentation/Assignment/ etc**	**ESE**
Weightage (%)	**20**	**30**	**50**

Relationship between the Course Outcomes (COs), Program Outcomes (POs) and Program Specific Objectives(PSOs)

PO/ CO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PS O1	PS O2	PS O3
CO1															
CO2															
CO3															
CO4															

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

Course Objectives

The student should be made to:
1. Learn to create and use a database
2. Be familiarized with a query language
3. Have hands on experience on DDL Commands
4. Have a good understanding of DML Commands and DCL commands
5. Familiarize advanced SQL queries.
6. Be exposed to different applications

Course Outcomes

At the end of the course, the student should be able to:
CO1. Design and implement a database schema for a given problem-domain

CO2. Populate and query a database
CO3. Create and maintain tables using PL/SQL.
CO4. Prepare reports.

List of Experiments:

1. Creation of a database and writing SQL queries to retrieve information from the database.
2. Performing Insertion, Deletion, Modifying, Altering, Updating and Viewing records based on conditions.
3. Creation of Views, Synonyms, Sequence, Indexes, Save point.
4. Creating an Employee database to set various constraints.
5. Creating relationship between the databases.
6. Study of PL/SQL block.
7. Write a PL/SQL block to satisfy some conditions by accepting input from the user.
8. Write a PL/SQL block that handles all types of exceptions.
9. Creation of Procedures.
10. Creation of database triggers and functions
11. Mini project (Application Development using Oracle/ Mysql)
a) Inventory Control System.
b) Material Requirement Processing.
c) Hospital Management System.
d) Railway Reservation System.
e) Personal Information System.
f) Web Based User Identification System.
g) Timetable Management System.
h) Hotel Management System

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme:

Components	MSE	Presentation/Assignment/ etc	ESE
Weightage (%)	20%	30%	50%

Relationship between the Course Outcomes (COs), Program Outcomes (POs) and Program Specific Objectives (PSOs)

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PSO1	PSO2
CO1												
CO2												
CO3												
CO4												
CO5												
Avg												

1= Weak 2= Moderate 3= Strong

	Soft Computing Lab	L	T	P	C
Version 1.0		0	0	0	
Pre-requisites/Exposure	• Proficiency with algorithms. • Programming skills in C, C++, or Java, MATLAB, etc.				
Co-requisites	--				

Course Objectives

The Course objective is to provide overview on Soft Computing concepts, technologies, and applications. Upon the completion of this course, the students will be having the knowledge on the underlying principle of soft computing with its usage in various application and different soft computing tools to solve real life problems

Course Outcomes

On completion of this course, the students will be able to
CO1: Classify the various soft computing frameworks.
CO2: Be familiar with the design of neural networks, fuzzy logic and fuzzy systems
CO3: Develop application on different soft computing techniques like Fuzzy, GA and Neural network
CO4: Implement Neuro-Fuzzy and Neuro-Fuzz-GA expert system
CO5: Learn mathematical background for optimized genetic programming

Catalog Description

Soft computing is an emerging approach to computing which parallel the remarkable ability of the human mind to reason and learn in an environment of uncertainty and imprecision. Soft computing is based on some biological inspired methodologies such as genetics, evolution, ant's behaviors, particles swarming, human nervous systems, etc. Now, soft computing is the only solution when we don't have any mathematical modelling of problem solving (i.e., algorithm), need a solution to a complex problem in real time, easy to adapt with changed scenario and can be implemented with parallel computing. It has enormous applications in many application areas such as medical diagnosis, computer vision, hand written character recondition, pattern recognition, machine intelligence, weather forecasting, network optimization, VLSI design, etc.

List of Experiments:

Experiment 01 - Create a perceptron with appropriate no. of inputs and outputs. Train it using fixed increment learning algorithm until no change in weights is required. Output the final weights.

Experiment 02 - Create a simple ADALINE network with appropriate no. of input and output nodes. Train it using delta learning rule until no change in weights is required. Output the final weights.

Experiment 03 - Train the autocorrelator by given patterns: A1=(-1,1,-1,1), A2=(1,1,1,-1), A3=(-1, -1, - 1, 1). Test it using patterns: Ax=(-1,1,-1,1), Ay=(1,1,1,1), Az=(-1,-1,-1,-1).

Experiment 04 - Train the hetrocorrelator using multiple training encoding strategy for given patterns: A1=(000111001) B1=(010000111), A2=(111001110) B2=(100000001), A3=(110110101) B3(101001010). Test it using pattern A2.

Experiment 05 - Implement Union, Intersection, Complement and Difference operations on fuzzy sets. Also create fuzzy relation by Cartesian product of any two fuzzy sets and perform max- min composition on any two fuzzy relations.

Experiment 06 - Solve Greg Viot's fuzzy cruise controller using MATLAB Fuzzy logic toolbox.

Experiment 07 - Solve Air Conditioner Controller using MATLAB Fuzzy logic toolbox.

Experiment 08 - Implement TSP using GA.

Textbooks

1. R. Rajasekaran and G. A and Vijayalakshmi Pa, *Neural Networks, Fuzzy Logic, and Genetic Algorithms: Synthesis and Applications*, Prentice Hall of India
2. D. E. Goldberg, *Genetic Algorithms in Search, Optimisation, and Machine Learning*, Addison-Wesley

Reference Books

1. L. Fausett, *Fundamentals of Neural Networks*, Prentice Hall
2. T. Ross, *Fuzzy Logic with Engineering Applications*, Tata McGraw Hill

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination
Examination Scheme:

Components of evaluation

1. Viva voce / Quiz (50%) + Performance & Records (50%).
2. Lab performance and record evaluation shall be a continuous process throughout the semester.
3. Minimum three Viva voce/ Quiz based on practical sessions shall be conducted during the semester.

Relationship between the Course Outcomes (COs), Program Outcomes (POs) and Program specific Outcomes (PSOs)

	PO1	PO2	PO3	PO 4	PO 5	PO6	PO 7	PO8	PO9	PO 10	PSO 1	PSO 2
CO1				2							2	1

CO2				2							2	1
CO3				2							2	1

1 = weak 2 = moderate 3 = strong

	JAVA programming Lab	L	T	P	C
Version 1.0		0	0	2	2
Pre-requisites/Exposure	Basic Knowledge of C/C++ Programming. Basic Knowledge of Object Oriented Design.				
Co-requisites	Knowledge of Computer Programming				

Course Objectives

The objectives of this course are to:

1. Create Java programs that leverage the object-oriented features.
2. Design & implement multithreading and data structure.
3. Learn the concepts of JDBC and Servlets.

Course Outcomes

On completion of this course, the students will be able to

CO1. Understand complete architecture of Java Platform and able to relate with OOPS concepts
CO2. Analyze real world object-oriented concepts and incorporate into the Java programming Language.
CO3. Implement and execute programs which are based on the OOPS concept.
CO4. Differentiate Java Programming from other OOPS based programming languages.
CO5. Develop basic level applications based on core java and advance java skills.

Catalog Description

Java is a programming language and computing platform, first released by Sun Microsystems in 1995. Many of applications and websites will not work unless you have Java installed, and more are creating every day. Java is fast, secure, and reliable. From laptops to datacentres, game consoles to scientific supercomputers, cell phones to the Internet, Java is everywhere.This Java Programming lab training covers the core language features and Application Programming Interfaces (API) you will use to design object-oriented applications with Java Standard Edition (Java SE) Platform.

List of Experiments

Experiment 01 - Introduction to Java Environment : This includes installation of JDK ,Setting of path and classpath and Introduction to Eclipse
Experiment 02 - Basic Java Programming: To develop simple java programs using various control structures.

Experiment 03 - Classes and Objects in PHP: To develop Java programs to understand the concepts of classes and objects.

Experiment 04 – Inheritance : To work with the various types of Inheritance.
Experiment 05 - Packages: To work with various packages in JAVA.
Experiment 06 – Abstraction in Java: To work with the abstraction concepts like abstract class and interfaces.
Experiment 07 - Exception Handling: To develop simple web programs with exception handling
Experiment 08- Multithreading: To work with the concepts to implement multithreading.

Experiment 09 - Database Connectivity: To develop simple programs to connect the Java with the database.
Experiment 10- Servlet: To work with the server side servlet technologies.

Text Books

1. The Java Programming Language 3rd Edition, Ken Arnold, James Gosling, Pearson
2. A premier guide to SCJP 3rd Edition, Khalid Mughal, Pearson
3. Thinking in Java, 3rd Edition, Bruce Ackel, Pearson
4. Video resources http://www.youtube.com and blackboard

Continuous Evaluation- There will be continuous evaluation for all practical subjects of SCS during the semester. The performance of a student in a Practical subject will be evaluated as per process given below:

Components of evaluation

1. Viva voce / Quiz (50%) + Performance & Records (50%).
2. Lab performance and record evaluation shall be a continuous process throughout the semester.
3. Minimum three Viva voce/ Quiz based on practical sessions shall be conducted during the semester.

	Object Oriented Analysis and Design using UML lab	L	T	P	C
Version 1.0		0	0	1	1
Pre-requisites/Exposure	--				
Co-requisites	--				

Course Objectives

1. The student should be able to design different UML diagrams like Use case diagram, class diagram and sequence diagrams.
2. To select and apply in autonomy appropriate technologies and techniques for different types of testing all over the software process development in different domains and contexts.

Course Outcomes

On completion of this course, the students will be able to

CO1. State the importance of object oriented modeling with UML.
CO2. Identify the role of actors, classes and objects in a software design.
CO3. Demonstrate the models for activity relationship and interaction modeling.
CO4. Analyze events and design state diagram.
CO5. Evaluate component and deployment diagrams.

List of Experiments

Experiment 1

OOAD Overview, UML OVERVIEW, Goals of UML, A conceptual model of UML, Object oriented concepts, Role of UML in OO design.

Experiment 2

Give overview of different UML tools available. Introduce Star UML. Also, take some examples to let students find entities, relationships

Experiment 3

Explain Use Case Diagram.(Use case ,Actors, Relationships)
Draw Use Case Diagram for following
a) An organization wants to develop the system based on following Requirements:
1. Students have to register first through fill up the registration form.

2. Faculty can prepare the grade sheet of concern subjects.
3. Student and faculty can see the result of each student.
4. Faculty can generate the report about registered students.
5. System can verify the students Registration Details First.
6. Student may have to fill up the hostel details during registration process.

b) Consider a book store in a shopping mall. The customer selects the books from racks to purchase. The customer brings selected books to cashier. The cashier scans each item with checkout system to prepare an order. The cashier requests to customer for payment. The customer gives credit card to cashier. The verifier and checkout system scans the card. The verifier accepts the card and payment is accepted. Customer signs the credit card slip. The purchased books are handed over to customer.

Experiment 4

Give overview of class diagram and draw a class diagram online ticket reservation system. Online ticket reservation system for railway department has to be developed.

The System developed should contain following features

1. The System should provide information about arrival and departure trains along with information about stations through which it passes.
2. Search about train passing through stations can be obtained either by means of train no, train name or specifying the source and destination stations.
3. While displaying information about train it has provide following information's
a) Stations through which train passes along with arrival and departure time.
b) Availability of seats in different classes along with waiting list.

4. While reserving ticket online the system obtain following information's from the user
a) Passenger name, Gender, Age, Address
b) Credit Card No, Bank Name
c) Class through passenger is going to travel i.e First class or Second class or AC
d) Train no and Train name, Date of Journey and number of tickets to be booked.

5. Based on the availability of tickets the ticket has to be issued. The ticked issued should contain the following information's PNR NO, Train No, Date, K.M., no of adults and children, Ticket No, Class, Ticket No, Coach, Seat/Berth, Sex, Age, Reservation fee, Total Cash, Train Name, Departure time.

6. Cancellation of booked tickets should be available.

Experiment 5

Draw detailed Activity diagram of Hospital Management System for the following:

1. System helps in registering information about patients and handles patient's query.

2. A unique ID is generated for each patient after registration for maintains medical history of patient.
3. System also monitors the doctor appointments, when the ID is generated the patient receives the appointment time and number from the receptionist and accordingly visits the doctor.

4. This system also deals with testing appointments as and when ID is generated the patient receives the appointment time and number and accordingly undergoes the test.

5. It also deals with bed allotments to various patients by checking their ID. It also undergoes various operations by diagnosing the patients.

6. The system identifies whether the person is a doctor or staff and handles various activities such as draw salary and gives salary, also it adds doctor/staff information into database.

7. This system is responsible for handling various other activities like deleting, editing doctor/staff information into the database.

8. As per doctor diagnoses the patient, gives treatment and gives suggestions to patients and prescribe laboratory tests and medicines.

9. This system also takes care of medical equipment, doctor visit, vitals recording, patient case sheet, diet ordering, blood requisition, transfer information and discharge information, maintenance of wards, inter and intra wards transfers also it generates patient's discharge summary which includes patients health at the time of discharge, medical history, various diagnosis and drug prescriptions, history of patients illness and course in hospital.

10. Patient can pay bill through credit card, cash orcheque whose information is maintained by this system.

Experiment 6

Develop a system in terms of accessing ATM. To develop an ATM System for XYZ Bank The System developed should contain the following features

1. The Customer login into the system using Credit Card No or Debit Card no and Pin Number. The system checks for validation.

2. The System queries the customer for the type of account either S.B Account or Credit. After getting the type of account the system shows the amount left.

3. The System then queries the customer for required amount. The user enters the amount and gets the money.

Experiment 7

To develop a Library Management System.
The basic functional requirements are:
a) Users of the library are either normal or staff

b) A user can borrow a book
c) A staff person can borrow a book
d) The library contains one million books
e) If a user asks a book that has been borrowed, her request is inserted in a waiting list
f) Staff has no priority in borrowing books

Experiment 8
Prepare a system for course registration of your branch

As the Head of Centre for Information Technology for UPES you are tasked with developing a new students registration system. The college would like a new client-server system to replace its much older system. The new system will allow students to register for courses and view report cards from personal computers attached to the campus LAN. Professors will be able to access the system. To sign up to teach courses as well as record grades.

At the beginning of each semester, students may request a course catalogue containing list course offerings for the semester. Information about each course, such as professor, department and prerequisites, will be included to help students make information decisions.
The new system will allow students to select four course offerings for the coming semester. Course offerings will have a maximum of ten students and minimum of three students. A course offering with fewer than 3 students will be cancelled. If a course is cancelled, the students will be intimated and they will be requested to change their course choices.
At the end of the semester, the students will be able to access the system to view an electronic report card. Since student grades are sensitive information, the system must employ extra security measures to prevent unauthorized access. Professors must able to access the online system to indicate which courses will be teaching. They will need to see which students signed up for their course offerings. In addition, the professors will be able to record the Grades for the students in each class.

Text Books

1. Object-Oriented Modeling and Design with UML, Michael R Blaha& James R Rumbaugh, Pearson Education
2. Object Oriented Analysis and Design, AtulKahate, TMH
3. Applying UML and Patterns, Craig Larmen, Pearson Education
4. The Unified Modeling Language User Guide, Grady Booch&James Rumbaugh&Ivar Jacobson, Addison Wesley

Continuous Evaluation- There will be continuous evaluation for all practical subjects of SCS during the semester. The performance of a student in a Practical subject will be evaluated as per process given below:

Components of evaluation

d. Viva voce / Quiz (50%) + Performance & Records (50%).

e. Lab performance and record evaluation shall be a continuous process throughout the semester.
f. Minimum three Viva voce/ Quiz based on practical sessions shall be conducted during the semester.

Relationship between the Course Outcomes (COs) and Program Outcomes (POs) and Program Specific Outcomes (PSOs)

Course Outcomes	PO1	PO2	PO3	PO 4	PO 5	PO6	PO 7	PO8	PO9	PO 10	PSO 1	PSO2
CO1												
CO2												
CO3												
CO4												
Average												

1=Weak 2=Moderate 3=Strong

SEMESTER III

	Modeling and Simulation	L	T	P	C
Version 1.0		3	0	0	3
Pre-requisites/Exposure	Object Oriented Modelling and Design				
Co-requisites					

Course Objectives

The objectives of this course are to:

1. To understand the techniques of modeling in the context of hierarchy of knowledge about a system and develop the capability to apply the same to study systems through available software.
2. To learn different types of simulation techniques.
3. To simulate the models for the purpose of optimum control by using software.

Course Outcomes

On completion of this course, the students will be able to

CO1. Understand the basics of simulation modeling and replicating the practical situations in organizations
CO2. Evaluate statistical models in simulation.
CO3. Analysis of Simulation models using input analyzer, and output analyzer
CO4. Explain Verification and Validation of simulation model.

Catalog Description

Modeling and Simulation has become an essential tool for optimum design and the course aims to impart an overview of the modeling and simulation approaches with emphasis on applications of modelling and Simulation.

Course Content

Unit I: **[5 lecture hours]**

INTRODUCTION

When simulation is the appropriate tool and when it is not appropriate; Advantages and disadvantages of Simulation; Areas of application; Systems and system environment; Components of a system; Discrete and continuous systems; Model of a system; Types of Models; Discrete-Event System Simulation; Steps in a Simulation Study.

Unit II: **[6 lecture hours]**

STATISTICAL MODELS IN SIMULATION

Review of terminology and concepts; Random Variables, Probability Distribution ,Probability distribution function, Useful statistical models; Discrete distributions; Continuous distributions; Poisson process; Empirical distributions.

Unit III: **[6 lecture hours]**

GENERAL PRINCIPLES OF SIMULATION SOFTWARE

Concepts in Discrete-Event Simulation: The Event-Scheduling / Time-Advance Algorithm, World Views, Manual simulation Using Event Scheduling; List processing. Simulation in Java

Unit IV:

INPUT MODELING **[6 lecture hours]**

Data Collection; Identifying the distribution with data; Parameter estimation; Goodness ofFit Tests; Fitting a non-stationary Poisson process; Selecting input models withoutdata; Multivariate and Time-Series input models, uniformity and independence, Chi-Squaretest, K-S Test.

Unit V: **[6 lecture hours]**

VERIFICATION AND VALIDATION

Verification, Calibration, and Validation; Optimization: Model building, verification and validation; Verification of simulation models; Calibration and validation of models

Unit VI: **[7 lecture hours]**

ESTIMATION OF ABSOLUTE PERFORMANCE & COMPUTER SYSTEM SIMULATION

S Types of simulations with respect to output analysis; Stochastic nature of output data; Absolute measures of performance and their estimation; Output analysis for terminating simulations; Output analysis for steady-state simulations.

Text Books

T1: Introduction to Computer Simulation: A System Dynamics Modeling Approach: Nancy Roberts.
T2: Modeling and Simulation by Hartmut Bossel

Reference Books

R1: Zeigler, B. P., Muzy, A., & Kofman, E. (2018). Theory of Modeling and Simulation: Discrete Event & Iterative System Computational Foundations. Academic Press.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination
Examination Scheme:

Components	MSE	Presentation/Assignment/ etc	ESE
Weightage (%)	20	30	50

Relationship between the Course Outcomes (COs), Program Outcomes (POs) and Program Specific Objectives(PSOs)

PO/ CO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PS O1	PS O2	PS O3
CO1															
CO2															
CO3															
CO4															

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

	Computer Networks	L	T	P	C
Version 1.0		3	0	0	3
Pre-Requisites/Exposure	Basic knowledge of computer systems and data structures				
Co-Requisites	Basic Mathematics				

Course Objectives

To help in understanding the concepts of communications and computer networks.

Course Outcomes

On completion of this course, the students will be able to

CO1. Discuss concepts of data communication and layered model of networks.

CO2. Describe concepts of flow and error control in networks.

CO3: Demonstrate various addressing schemes and routing strategies.

CO4: Describe various functionalities at transport and application layer.

Catalog Description

This course provides an introduction to data communication and computer networks. The course covers the principles of data communication, the fundamentals of signaling, basic transmission concepts, transmission media, circuit control, line sharing techniques, physical and data link layer protocols, error detection and correction, data compression, common carrier services and data networks. Various routing strategies, functions of protocols included in TCP/IP protocol suite, different congestion control mechanisms and the protocols used at application layers (like HTTP, FTP, SNMP, SMTP etc.) have also been included in the course.

Course Content

UNIT I:

Data Communication **04 Lecture Hours**

Introduction, Theoretical Model for Communication, Bandwidth, throughput, Analog and Digital Data Transmission, Transmission Impairments-Guided transmission media- Wireless Transmission- Line-of-sight Transmission, data rate Channel Capacity-Band width and Shannon's capacity equation, Digital Data Communication Techniques: Asynchronous and Synchronous Transmission, Concept of circuit, Message, Packet Switching with their timing diagram, Comparison of Switching Techniques.

UNIT II:

Physical Layer **05 Lecture Hours**

Evolution of computer network, Layered Network Architecture, OSI Layer Model, TCP/IP, ATM, three tier architecture, System Network architecture, Protocols & Standardization, Transmission media, Topology, Line Discipline , ISDN, Frame relay, Ethernet switches, Fast Ethernet and Gigabit Ethernet, FDDI.

UNIT III

Data Link Layer **06 Lecture Hours**

Data link layer services: Error detect and correction techniques, Elementary Data link layer protocols, sliding window protocols, stop and wait protocol, selective repeat, HDLC ,Multiple access protocols, TDM, FDM, CDMA Random access protocols: ALOHA, CSMA,CSMA/CD,CSMA/CA. IEEE 802 standards for LAN & WAN: 802.3, 802.4, 802.5, 802.6, 802.2 & their comparison.

UNIT IV Network Layer **03 Lecture Hours**

Network layer Services, Datagram and Virtual circuit services, IP datagram format and Types of Services, Datagram encapsulation and Fragmentation, Reassembly and fragmentation. Ip addressing, subnetting and supernetting.

UNIT V: Routing Techniques 07 Lecture Hours

Routing: Link state routing, distant vector routing, hierarchical routing, multicast routing. Internet Routing Between Peers (BGP)-Routing Within An Autonomous System (RIP, OSPF). IPv4, IPv6: Frame formats-Comparison with IPv4. Introduction to ICMP, DHCP and NAT. Network Management: SNMP and RMON models

UNIT VI: Transport Layer And Application Layer 09 Lecture Hours

Transport Layer Services, Relationship with Network Layer, Relationship with Application Layer, Multiplexing and De multiplexing, UDP, TCP: Header ,Segment Structure, Services, Connection establishment and termination, Flow control and window size advertising, TCP time out and re-transmission, Congestion Control, TCP Fairness, Delay Modeling. Application layer protocols:- WWW and HTTP, FTP, DNS, SMTP, SNMP, RPC, P2P File sharing, Domain Name system (DNS)

Text Books

1. Youlu Zheng and Shakil Akhtar, Networks for Computer Scientist and Engineers, Oxford
2. University Press,2006
3. Behrouz A. Fourouzan ,Data Communications and Networking, 2/e Tat McGrawhill,2000
4. James F. Kurose and Keith W. Ross, Computer Networking – A Top-Down Approach
5. Featuring the Internet,2/e Pearson Education ,2003

Reference Books

1. S. Keshav, An Engineering Approach to Computer Networking, Pearson education ,2002
2. F. Halsall, Data Communication, Computer Networks and Open Systems, Addison Wesley, 1996
3. Andrew S. Tanenbaum, Computer Networks , 4/e, Pearson education, 2003
4. Leon-Garcia and I. Widjaja, Communication Networks, Tata McGraw Hill, 2000
5. Bertsekas and Gallagar , Data Networks, 2/e, PHI, 1992

6. Douglas E Comer ,Computer Networks and Internet's, 2/e Pearson Education,2004
7. Gallo, Computer Communication and Networking Technologies, Thomson Learning

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme:

Components	Internal	Mid Term	ESE	Total
Weightage (%)	30%	20%	50%	100%

Relationship between the Course Outcomes (COs), Program Outcomes (POs) and Program Specific Objectives(PSOs)

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1															
CO2															
CO3															
CO4															
Average															

1=weak 2= moderate 3=strong

	Computer Graphics	L	T	P	C
Version 1.0		3	0	0	3
Pre-requisites/Exposure	• Basic knowledge about matrix and geometry. • Knowledge of C/C++				
Co-requisites	--				

Course Objectives

1. Learn to create 2D and 3D objects.
2. Able to apply various transformations on the 2D and 3D objects.
3. To apply hidden surface removal techniques along with various shading algorithms
4. Create 3D graphics with realistic effects

Course Outcomes

On completion of this course, the students will be able to

CO1. Classify various graphics hardware and software devices.

CO2. Use primitive operations to create 2D and 3D objects and perform various operations thereon.

CO3. Perform complex 2D and 3D transformations on objects.

CO4. Implement various hidden surface removal techniques.

CO5. Create 3D realistic imagery by applying shading and colouring techniques on objects.

Catalog Description

This course discusses the theory behind computer graphics and includes many computer graphics algorithms. It is a study of the hardware and software principles of interactive raster graphics. Topics include an introduction to the basic concepts, 2-D and 3-D modeling and transformations, viewing transformations, projections, rendering techniques, graphical software packages and graphics systems. Students will use standard graphics application programming interface (OpenGL) to reinforce concepts and study fundamental computer graphics. The course also includes vertex processing; lighting and shading; rasterization including line and polygon drawing; ray casting; ray tracing, computer graphics in games visualization.

Course Content

UNIT I: **5 Lecture Hours**

Introduction to Computer Graphics and openGL:

Overview of Computer Graphics, Computer Graphics Application and Software, Description of some graphics devices, Introduction to pixel. Why OpenGL, Features in OpenGL, OpenGL operations, Abstractions in OpenGL – GL, GLU & GLUT, 3D viewing pipeline, viewing matrix specifications, a few examples and demos of OpenGL programs.

UNIT II: **8 Lecture Hours**

Scan conversion – lines, circles and Ellipses; Filling polygons and clipping algorithms

Scan Converting Lines, Mid-point criteria, Aliasing and Antialiasing ,Problems of Aliasing, end-point ordering and clipping lines, Scan Converting Circles, Scan Converting Ellipses, Filling Polygons Clipping Lines algorithms– Cyrus Beck, Cohen-Sutherland and Liang-Barsky, Clipping Polygons, problem with multiple components.

UNIT III **10 Lecture Hours**

2-D and 3-D Transformations

Transformations and Matrices, Transformation Conventions, 2D Transformations, Homogeneous Coordinates and Matrix Representation of 2D Transformations, Translations and Homogeneous Coordinates, Rotation, Reflection, Scaling, Combined Transformation, Transformation of Points, Transformation of The UNIT Square, Rotation About an Arbitrary Point, Reflection through an Arbitrary Line, A Geometric Interpretation of Homogeneous Coordinates, The Window-to-Viewport Transformations.

Three-Dimensional Scaling, Three-Dimensional Shearing, Three-Dimensional Rotation, Three-Dimensional Reflection, Three-Dimensional Translation, Multiple Transformation, Rotation about an Arbitrary Axis in Space, Reflection through an Arbitrary Plane, Matrix Representation of 3D Transformations, Composition of 3D Transformations.

UNIT IV **8 Lecture Hours**

Rendering

Visible-Surface Determination, Techniques for efficient Visible-Surface Algorithms, Categories of algorithms, Back face removal, The z-Buffer Algorithm, Scan-line method, Painter's algorithms (depth sorting), Area sub-division method, BSP trees, Visible-Surface Ray Tracing, comparison of the methods.

Illumination and Shading Models for Polygons, Reflectance properties of surfaces, Ambient, Specular and Diffuse reflections, Phong's model, Gouraud shading, some examples.

UNIT V **5 Lecture Hours**

Plane Curves and Surfaces

Curve Representation, Nonparametric Curves, Cubic Splines, , Bezier Curves, Bspline Curves, B-spline Curve Fit, B-spline Curve Subdivision, Parametric Cubic Curves, Quadric Surfaces. Bezier Surfaces, Fractals.

Text Books

1. Donald D. Hearn and M. Pauline Baker(2014), Computer Graphics, Pearson Education India, ISBN: 9788177587654.

Reference Books

1. David F. Rogers and J. Alan Adams, Mathematical Elements for Computer Graphics, McGraw-
Hill, Inc

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme:

Components	Internal	Mid Term	ESE
Weightage (%)	30%	20%	50%

Relationship between the Course Outcomes (COs), Program Outcomes (POs) and Program Specific Objectives(PSOs)

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1															
CO2															
CO3															
CO4															
CO5															
Average															

1=weak 2= moderate 3=strong

	Deep Learning and ANN	L	T	P	C
Version 1.0		3	0	0	3
Pre-requisites/Exposure	Working knowledge of Linear Algebra, Probability Theory, statistics				
Co-requisites	--				

Course Objectives

1. To learn the fundamentals of deep learning.
2. To grasp the contemporary research in many closely related fields, including computer vision, natural language processing, speech recognition and robotics.

Course Outcomes

On completion of this course, the students will be able to

CO1. Describe the concepts of deep learning.
CO2. Discuss basic techniques and modern approaches in designing, training and visualizing feedforward neural network architectures, convolutional neural networks.
CO3. Apply the deep learning concepts to many closely related fields, including computer vision, natural language processing, speech recognition and robotics

Catalog Description

Deep learning is a sub-field of machine learning that focuses on learning complex, hierarchical feature representations from raw data. This course comprises the fundamental principles, underlying mathematics, and implementation details of deep learning. This includes the concepts and methods used to optimize these highly parameterized models (gradient descent and backpropagation, and more generally computation graphs), the modules that make them up (linear, convolution, and pooling layers, activation functions, etc.), and common neural network architectures (convolutional neural networks, recurrent neural networks, etc.).

Course Content

Unit 1: Introduction **6 lecture hours**

How is deep learning different from another machine learning? AI vs ML vs DL, Deep learning capabilities, Other approaches to artificial intelligence, what is special about deep learning? Relevance of deep learning.

Unit 2: Diving to the Depths of Deep Learning **9 lecture hours**

Deep learning depths, Model: The molecules of DL, Loss functions in neural networks, Optimizers in neural networks, Activation functions, Finding the perfect fit, Running deep learning

algorithms: The frameworks, Real examples and actual schematics of building neural nets, Data preparation and label preparation, Examples of neural networks at work, Readying data for neural nets, Constructing the network, ReLU, Constructing the network, Approach validation, Plotting the loss from validation & training, What experiments do we run next? An example in regression: Guess the price of the house, Processing the data, Building the network, K-fold approach for validating algorithm, K-fold approach: In code.

Unit 3: Advanced Topics 9 **lecture hours**

Convolutional neural networks- Fundamentals, architectures, pooling, visualization, deep learning for spatial localization. Recurrent neural networks- long-short term memory (LSTM), language models, machine translation, image captioning, video processing, visual question answering, video processing, learning from descriptions. Deep generative models- Auto-encoders, variational auto-encoders, generative adversarial networks, autoregressive models, generative image models, unsupervised and self-supervised representation learning Deep reinforcement learning- Policy gradient methods, Q-Learning

Unit 4. Tangential Topics of Machine Learning and Neural Networks 11 lecture hours

Information-theoretic machine learning, Basic concepts of information theory, Information processing in analog channels, Some common terminologies, Deep neural networks: Information theoretical perspective, Information bottleneck methodology, Concepts to remember, Capacity modelling theorems, Characteristics of deep neural network layers, Phases in double optimization, Encoder and decoder in DNN, Learning theory, Stochastic relation and hidden layers, Applying information gain, Hebbian learning, Implementation of Hebbian learning in a perceptron, Limitations of Hebbian learning, Competitive learning, What is competition in neural networks? Characteristics of competitive learning, Criteria for competitive learning, Architecture and implementation, Mathematical representation, Competitive learning, Competitive learning rule, Hebbian learning and competitive learning, Boltzmann learning, Boltzmann machines, Energy-Based Models (EBMs), Restricted Boltzmann machines, Restricted Boltzmann machines: Working, Radial-basis function networks, RBF network architecture, RBF neuron activation function, RBFN as a neural network, Advantages of using RBNN than the MLP.

Text Books

1. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, 2016. http://www. deeplearningbook.org.
2. Handbook of Simulation: Principles, Methodology, Advances, Applications, and Practice Jerry Banks (Editor).

References

1. K. P. Murphy, Machine Learning: A Probabilistic Perspective, MIT Press, 2012.D Srinivasan and R Gopalswamy; “Software Testing: Principles and Practices”, Pearson Education, 2006.

2. C. M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination
Examination Scheme:

Components	MSE	Presentation/Assignment/ etc	ESE
Weightage (%)	20%	30%	50%

Relationship between the Course Outcomes (COs) and Program Outcomes (POs) and Program Specific Outcomes (PSOs)

Course Outcomes	PO1	PO2	PO3	PO 4	PO 5	PO6	PO 7	PO8	PO9	PO 10	PSO 1	PSO2
CO1												
CO2												
CO3												
CO4												
Average												

1=Weak 2=Moderate 3=Strong

	Natural Language Processing	L	T	P	C
Version 1.0		3	0	0	3
Pre-requisites/Exposure	Basics of Probability, Basics of Automata Theory				
Co-requisites	Python Programming				

Course Objectives

The objectives of this course are to

1. To introduce the concept of Natural Language Understanding & Natural Language Generation.
2. To develop the concept of statistical and probabilistic approach of language modelling.
3. To extend the knowledge of parser by parsing the natural language sentences.
4. To enrich the knowledge with different corpuses and different tools being used for machine translation.
5. To comprehend the application area of NLP.

Course Outcomes

On completion of this course, the students will be able to

CO1: Understand the techniques in NLP.
CO2. Analyse and apply the natural language generation.
CO3. Study and Implementation of machine translation.
CO4: Analyse and apply the information retrieval techniques.
CO5: Exploration of different application of NLP

Catalog Description

This course starts with the basics of text processing including basic pre-processing, spelling correction, language modeling, Part-of-Speech tagging, Constituency and Dependency Parsing, Lexical Semantics, distributional Semantics and topic models. Finally, the course also covers some of the most interesting applications of text mining such as entity linking, relation extraction, text summarization, text classification, sentiment analysis and opinion mining.

Course Content

Unit 1. Introduction

Natural language, definition, processing, NLU, NLG, Classical Approach of NLP, Formal language, Linguistic and Language Processing, Steps of NLP, Why NLP is hard? Morphological

and lexical analysis, Syntactic analysis, Semantic analysis, Discourse integration, Pragmatic analysis, Corpus Creation

Unit 2. Language Modelling

Introduction of Language modelling, Markov Models, Bigram, Tri-gram Language Models, Evaluation of Language Models: Maximum likelihood estimation, Perplexity, Smoothing, Parts-of-speech Tagging Problems: Hidden Markov Model, Viterbi Algorithm

Unit 3. Semantic and Syntactic Processing

Introduction to Semantics and Knowledge representation, Introduction to Grammar and Parsers: Probabilistic Context Free Grammar, CKY Parsing Algorithm, Introduction of Syntactic Processing: Meaning elaboration on contextual and world knowledge, Garden Path Sentences, Probability based syntactic processing, Word Sense Disambiguity, Co-reference Resolution.

Unit 4. Applications

Information Retrieval, Question Answering, Biomedical Text Extraction, Sentiment Analysis, Text Summarization, Document Clustering and Automated Chatbot,

Unit 5. Machine Translation

Introduction of Machine Translation, Rules & Probability based Machine Translation, Case Study: Different tools for Machine Translation, Database Interface

SUGGESTED READINGS:

Text Books

1. Speech and Language Processing "An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition" Second Edition Daniel Jurafsky Stanford University James H.Martin University of Colorado at Boulder.

Reference Books:

1. "Natural Language Processing using Python" by Steve Bird et.al. O'Reilly Publication.

2. "Information Retrieval: Algorithms and Heuristics" by David A. Grossman, Ophir Frieder, Academic Press.
3. Natural Language Processing A paninian perspective, Prentice Hall, New Delhi, 1994 Akshar Bharati, Vineet Chaitanya, and Rajeev Sangal.

Web References:

1. https://onlinecourses-archive.nptel.ac.in/noc17_cs03/preview
2. http://www-nlp.stanford.edu/software/
3. www.cs.columbia.edu/~mcollins

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination
Examination Scheme:

Components	MSE	Quiz/Assignment/ etc.	ESE
Weightage (%)	20%	30%	50%

Relationship between the Course Outcomes (COs), Program Outcomes (POs) and Program Specific Objectives (PSOs)

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1															
CO2															
CO3															
CO4															
CO5															
Avg.															

1=weak 2= moderate 3=strong

	Game theory and Heuristics	L	T	P	C
Version 1.0		3	0	0	3
Pre-requisites/Exposure					
Co-requisites					

Course Objectives

1. To explain and predict how individuals behave in a specific strategic situation, and therefore help improve decision making
2. To explain in depth the standard equilibrium concepts in Game Theory
3. To illustrate the concepts, real-world examples and case studies
4. To design Repeated Games with public information
5. To design static and Dynamic games with incomplete information

Course Outcomes

On completion of this course, the students will be able to

CO1. Describe the strategic situations and represent them as games.
CO2. Solve simple games using various techniques.
CO3. Recommend and apply strategies to implement
CO4. Develop Static and Dynamic Games

Catalog Description

Course Content

Unit 1: Introduction to game theory **6 lecture hours**

Games and solutions - Game theory and mechanism design - Examples from networks - Strategic form games - Matrix and continuous games - Iterated strict dominance - Rationalizability - Nash Equilibrium - existence and uniqueness - Mixed and correlated equilibrium – Super modular games - Potential/congestion games – Existence and Properties of Nash Equilibria.

Unit 2: Extensive-Form Games **9 lecture hours**

Definition - Strategies and Equilibria in Extensive Form Games - Backward Induction and Subgame Perfection and its Critiques.

Unit 3: Repeated Games 9 **lecture hours**

Infinitely/finitely repeated games - Pareto Perfection and Renegotiation - Proofness in Repeated Games - Repeated Games with incomplete Public Information - Trigger strategies - Fork Theorem with Imperfect Public Information

Unit 4. Static Games with incomplete information **7 lecture hours**

Mixed and Behavioral strategies - Bayesian Nash equilibrium - Applications in auctions - Different auction formats - Revenue and efficiency properties of different auctions - Bayesian Games and Mechanism Design Principle - Single Agent - Several Agents - Further topics in Mechanism Design.

Unit 5 Dynamic Games with incomplete information **6 lecture hours**

Introduction - Perfect Bayesian Equilibrium in Multi-stage games - Extensive-Form and Strategic-Form Refinements - Reputation Effects - Sequential Bargaining under Incomplete Information.

Text Books

Fudenberg, Drew, Jean Tirole, "Game Theory", Cambridge, MA: MIT Press, 1991

References

1. Nisan, Noam, Tim Roughgarden, Eva Tardos, Vijay V. Vazirani, "Algorithmic Game Theory", Cambridge,UK: Cambridge University Press, 2007
2. Fudenberg, Drew, David Levine, "Theory of Learning in Games", Cambridge, MA: MIT Press, 1998.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination
Examination Scheme:

Components	MSE	Presentation/Assignment/ etc	ESE
Weightage (%)	20%	30%	50%

Relationship between the Course Outcomes (COs) and Program Outcomes (POs) and Program Specific Outcomes (PSOs)

Course Outcomes	PO1	PO2	PO3	PO 4	PO 5	PO6	PO 7	PO8	PO9	PO 10	PSO 1	PSO2
CO1												
CO2												
CO3												
CO4												
Average												

1=Weak 2=Moderate 3=Strong

	Information Retrieval	L	T	P	C
Version 1.0		3	0	0	3
Pre-requisites/Exposure					
Co-requisites					

Course Objectives

• To understand the basics of Information Retrieval and machine learning techniques for text classification and clustering.
•To understand various search engine system operations and recommender system.

Course Outcomes

On completion of this course, the students will be able to

CO1. Use an open source search engine framework and explore its capabilities.
CO2. Apply appropriate method of classification or clustering.
CO3. Design and implement innovative features in a search engine and recommender system

Catalog Description

This course includes basic concepts, tools & techniques in the field of Information Retrieval (IR) & Search. It will cover theoretical foundations, implementation aspects, issues and state-of-the-art in the area of information retrieval, representation, organization, indexing and categorization. The course will cover topics such as

Course Content

Unit 1: Introduction **6 lecture hours**

Information Retrieval – Early Developments – The IR Problem – The Users Task – Information versus Data Retrieval – The IR System – The Software Architecture of the IR System – The Retrieval and Ranking Processes – The Web – The e-Publishing Era – How the web changed Search – Practical Issues on the Web – How People Search – Search Interfaces Today – Visualization in Search Interfaces.

Unit 2: Modelling and retrieval Evaluation **9 lecture hours**

Basic IR Models – Boolean Model – TF-IDF (Term Frequency/Inverse Document Frequency) Weighting – Vector Model – Probabilistic Model – Latent Semantic Indexing Model – Neural

Network Model – Retrieval Evaluation – Retrieval Metrics – Precision and Recall – Reference Collection – User-based Evaluation – Relevance Feedback and Query Expansion – Explicit Relevance Feedback.

Unit 3: Text classification and clustering 9 **lecture hours**

A Characterization of Text Classification – Unsupervised Algorithms: Clustering – Naïve Text Classification – Supervised Algorithms – Decision Tree – k-NN Classifier – SVM Classifier – Feature Selection or Dimensionality Reduction – Evaluation metrics – Accuracy and Error – Organizing the classes – Indexing and Searching – Inverted Indexes – Sequential Searching – Multi-dimensional Indexing.

Unit 4. Web retrieval and web crawling **7 lecture hours**

The Web – Search Engine Architectures – Cluster based Architecture – Distributed Architectures – Search Engine Ranking – Link based Ranking – Simple Ranking Functions – Learning to Rank – Evaluations — Search Engine Ranking – Search Engine User Interaction – Browsing – Applications of a Web Crawler – Taxonomy – Architecture and Implementation – Scheduling Algorithms – Evaluation.

Unit 5 Recommender System **6 lecture hours**

Recommender Systems Functions – Data and Knowledge Sources – Recommendation Techniques – Basics of Content-based Recommender Systems – High Level Architecture – Advantages and Drawbacks of Content-based Filtering – Collaborative Filtering – Matrix factorization models – Neighborhood models.

Text Books

1. Ricardo Baeza-Yates and Berthier Ribeiro-Neto, ―Modern Information Retrieval: The Concepts and Technology behind Search, Second Edition, ACM Press Books, 2011.
2. Ricci, F, Rokach, L. Shapira, B.Kantor, ―Recommender Systems Handbook, First Edition, 2011.

References

1. C. Manning, P. Raghavan, and H. Schütze, ―Introduction to Information Retrieval, Cambridge University Press, 2008.

3. Stefan Buettcher, Charles L. A. Clarke and Gordon V. Cormack, ―Information Retrieval: Implementing and Evaluating Search Engines, The MIT Press, 2010.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination
Examination Scheme:

Components	MSE	Presentation/Assignment/ etc	ESE
Weightage (%)	20%	30%	50%

Relationship between the Course Outcomes (COs) and Program Outcomes (POs) and Program Specific Outcomes (PSOs)

Course Outcomes	PO1	PO2	PO3	PO 4	PO 5	PO6	PO 7	PO8	PO9	PO 10	PSO 1	PSO2
CO1												
CO2												
CO3												
CO4												
Average												

1=Weak 2=Moderate 3=Strong

	Image processing and Pattern Analysis	L	T	P	C
Version 1.0		**3**	**0**	**0**	**3**
Pre-requisites/Exposure	**AI**				
Co-requisites	**--**				

Course Objectives

The objective of this course is to imparts knowledge in the area of image and image processing, to understand fundamentals of digital image processing and to learn the fundamentals of Pattern recognition in order to choose an appropriate feature classification algorithm for pattern recognition problems and apply them properly using modern computing tools such as Matlab, Python, C/C++ etc.

Course Outcomes

On completion of this course, the students will be able to

1. understand the fundamental characteristics of digital systems used in imaging, together with general concepts of science, quantitative methods.
2. describe, analyze and reason about basic algorithms for image manipulation, characterization, filtering, segmentation, feature extraction and template matching in direct space and Fourier space.
3. possess advanced knowledge of how digital images are represented, manipulated, encoded and processed, with emphasis on algorithm design, implementation and performance evaluation.
4. learn the fundamentals of Pattern recognition and to choose appropriate feature classification algorithms for pattern recognition problems and apply them properly using modern computing tools such as Matlab, Python, C/C++ etc.

Course Description

This course is a graduate-level introductory course to the fundamentals of digital image processing and the theory of pattern analysis. It emphasizes general principles of image processing, rather than specific applications. Lectures will cover topics such as point operations; color processing, image thresholding/segmentation, morphological image processing, image filtering and DE convolution, noise reduction and restoration, scale-space techniques, feature extraction and recognition, image registration, and image matching. This course includes foundations of pattern recognition algorithms and machines, including statistical and structural methods. Data structures for pattern representation, feature discovery and selection, classification vs. description, parametric and non-parametric classification.

Course Content

UNIT-I: IMAGE PRELIMINARIES (2 L)
Frequency domain transformation techniques and their properties.
Image Acquisition: Energy, the optical system, image sensor and digital image formation. Gray scale and color images. analogue to digital conversion, image sampling and quantization, look-up table conversions, scaling, mathematical tools, digital image formats representation and description.

UNIT-II: IMAGE PROCESSING (8 L)
Image Point Processing: Gray-level mapping, non-liner graylevel mapping, image histogram, histogram stretching, histogram equalization, histogram matching, thresholding.
Neighborhood Processing: Median filter, mean filter, correlation, templet matching, edge detection and image sharpening.
Color image processing.
Morphology: Dilation & erosion, closing & opening and boundary detection

UNIT-III: IMAGE TRANSFORMATIONS (10 L)
Geometric transformations: Translation, rotation, scaling and shearing.
Frequency transformation: Discrete Fourier transform (DFT), fast Fourier transform (FFT) and short-time Fourier transform (STFT), 2-D Fourier transform, 1-D and 2-D filtering.
Multi-resolution Expansions: Pyramidal Multi-resolution analysis, Haar wavelet transforms in 1-D and 2-D, the fast wavelet transform, wavelet packets transform.

UNIT-IV: FEATURE EXTRACTION AND DIMENSION REDUCTION (12 L)
Color, Texture, Shape, Local Features, Spatial and frequency domain, HOG, Corner Detection, SIFT and SURF, Hough Transform, Principal Component Analysis

UNIT-V: PATTERN RECOGNITION (4 L)
The Unsupervised Clustering Algorithm, K-NN, Support Vector Machine, Neural Networks, Deep Learning-Overview

TEXT BOOKS

1. Digital Signal Processing- J G Proakis and D G Manolakis, Pearson, Fourth edition
2. Digital Image Processing – Ganzalez and Wood, Addison Wesley, 1993.
3. Fundamental of Image Processing – Anil K.Jain, Prentice Hall of India.
4. Pattern Classification – R.O. Duda, P.E. Hart and D.G. Stork, Second Edition John Wiley, 2006

REFERENCE BOOKS

1. Digital Picture Processing – Rosenfeld and Kak, vol.I & vol.II, Academic,1982

2. Computer Vision – Ballard and Brown, Prentice Hall, 1982
3. An Introduction to Digital Image Processing – Wayne Niblack, Prentice Hall, 1986
4. Pattern Recognition and Machine Learning – C. M. Bishop, Springer, 2009.
5. Pattern Recognition – S. Theodoridis and K. Koutroumbas, 4th Edition, Academic Press,2009

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme:

Components	MSE	Presentation/Assignment/ etc	ESE
Weightage (%)	20	30	50

Relationship between the Course Outcomes (COs), Program Outcomes (POs) and Program specific Outcomes (PSOs)

PO / CO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PS O1	PS O2	PS O3
CO 1															
CO 2															
CO 3															
CO 4															

1- Slight (low) 2- Moderate (Medium) 3-Substantial (high)

	Deep Learning and ANN lab	L	T	P	C
Version 1.0		**0**	**0**	**0**	**1**
Pre-requisites/Exposure	**-**				
Co-requisites	**--**				

Course Objectives

1. To implement the fundamentals of deep learning.
2. To grasp the contemporary research in many closely related fields, including computer vision, natural language processing, speech recognition and robotics.

Course Outcomes

On completion of this course, the students will be able to

CO1. Implement the deep learning concepts in programming setup, example networks, shallow and deep networks.

List of Experiments

Experiment 1

- Programming setup
- Tensorflow introduction
- Graph-based computation

Experiment 2

- Training Example Networks
- Perceptrons
- Shallow/Deep Networks

Experiment 3

@ eScience WRF Studio

- Working with Operators
- Designing various training procedures

- Projects discussion

Experiment 4

- Training CNN
- Examples from Computer Vision:
- Classification examples (AlexNet)
- Segmentation examples

Experiment 5

@ eScience WRF Studio

- Training RNNs
- Examples from NLP
- Examples from Robotics
- Project Pitches
- Experiment 6
  - Training GANs
  - Examples of Image Generation
- Experiment 7
  - Reinforcement Learning: Q-Learning
- Experiment 8
- Interactive Unsupervised Learning Lab

Continuous Evaluation- There will be continuous evaluation for all practical subjects of SCS during the semester. The performance of a student in a Practical subject will be evaluated as per process given below:

Components of evaluation

a. Viva voce / Quiz (50%) + Performance & Records (50%).
b. Lab performance and record evaluation shall be a continuous process throughout the semester.
c. Minimum three Viva voce/ Quiz based on practical sessions shall be conducted during the semester.

Relationship between the Course Outcomes (COs) and Program Outcomes (POs) and Program Specific Outcomes (PSOs)

Course Outcomes	PO1	PO2	PO3	PO 4	PO 5	PO6	PO 7	PO8	PO9	PO 10	PSO 1	PSO2
CO1												
CO2												
CO3												
CO4												
Average												

1=Weak 2=Moderate 3=Strong

	Domain Project	**L**	**T**	**P**	**C**
Version 1.0		**0**	**0**	**0**	**4**
Pre-requisites/Exposure	**-**				
Co-requisites	**--**				

SEMESTER IV

	Full time industry Project	L	T	P	C
Version 1.0		**0**	**0**	**0**	**20**
Pre-requisites/Exposure	**-**				
Co-requisites	**--**				