Andhra Pradesh State Council of Higher Education

B.Sc., Honours in Computer Science MAJOR

w.e.f AY 2023-24 onwards

COURSE STRUCTURE

Year	Semester	Course	Title	Hrs./ Week	Credits
I	I	1	Essentials and applications of Mathematical, Physical and Chemical Sciences	5	4
		2	Advances in Mathematical, Physical and Chemical Sciences	5	4
	II	3	Problem Solving using C - (T)	3	3
			Problem Solving using C- (P)	2	1
		4	Digital Logic Design- (T)	3	3
			Digital Logic Design- (P)	2	1
II	III	5	Object Oriented Programming using Java- (T)	3	3
			Object Oriented Programming using Java - (P)	2	1
		6	Data Structures using C - (T)	3	3
			Data Structures using C - (P)	2	1
		7	Computer Organization - (T)	3	3
			Computer Organization- (P)	2	1
		8	Operating Systems - (T)	3	3
			Operating Systems - (P)	2	1
	IV	9	Database Management System - (T)	3	3
			Database Management System - (P)	2	1
		10	Object Oriented Software Engineering - (T)	3	3
			Object Oriented Software Engineering - (P)	2	1
		11	Data Communications and Computer Networks - (T)	3	3
			Data Communications and Computer Networks - (P)	2	1
III	V	12	Web Interface Designing Technologies - (T)	3	3
			Web Interface Designing Technologies - (P)	2	1
		13	Web Applications Development using PHP & MYSQL - (T)	3	3
			Web Applications Development using PHP & MYSQL - (P)	2	1
		14 A	Internet of Things (T)	3	3
			Internet of Things (P)	2	1
			OR		
		14 B	Foundations of Data Science - (T)	3	3

			Foundations of Data Science - (P)	2	1
		15 A	IoT Applications Development and Programming - (T)	3	3
			IoT Applications Development and Programming - (P)	2	1
			OR		
		15 B	Application development using Python - (T)	3	3
			Application development using Python - (P)	2	1
	VI		Internship/ Apprenticeship		
IV	VII	16 A	Advanced Data Structures - (T)	3	3
			Advanced Data Structures - (P)	2	1
			OR		
		16 B	Artificial Intelligence - (T)	3	3
			Artificial Intelligence - (P)	2	1
		17 A	Computer Graphics - (T)	3	3
			Computer Graphics - (P)	2	1
			OR		
		17 B	Design and Analysis of Algorithms - (T)	3	3
			Design and Analysis of Algorithms - (P)	2	1
		18 A	Principles of Machine Learning - (T)	3	3
			Principles of Machine Learning - (P)	2	1
			OR		
		18 B	Software Testing- (T)	3	3
			Software Testing- (P)	2	1
	SEC	19 A	Advanced Java Programming - (T)	3	3
			Advanced Java Programming - (P)	2	1
			OR		
		19 B	Mobile Application Development - (T)	3	3
			Mobile Application Development - (P)	2	1
		20 A	MEAN Stack Development - (T)	3	3
			MEAN Stack Development - (P)	2	1
			OR		
		20 B	R Programming - (T)	3	3
			R Programming - (P)	2	1
	VIII	21 A	Big Data Technologies - (T)	3	3
			Big Data Technologies - (P)	2	1
			OR		
		21 B	Compiler Design - (T)	3	3
			Compiler Design - (P)	2	1
		22 A	Data Mining Concepts & Techniques - (T)	3	3
			Data Mining Concepts & Techniques - (P)	2	1
			OR		
		22 B	Digital Image Processing - (T)	3	3
			Digital Image Processing - (P)	2	1

		23A	Information Security and Cryptography - (T)	3	3
			Information Security and Cryptography - (P)	2	1
			OR		
		23 B	Mobile ADHOC and Sensor Networks - (T)	3	3
			Mobile ADHOC and Sensor Networks - (P)	2	1
	SEC	24 A	Advanced DBMS - (T)	3	3
			Advanced DBMS - (P)	2	1
			OR		
		24 B	Cloud Computing - (T)	3	3
			Cloud Computing - (P)	2	1
		25 A	Computer Vision - (T)	3	3
			Computer Vision - (P)	2	1
			OR		
		25 B	Digital Forensics - (T)	3	3
			Digital Forensics - (P)	2	1

I SEMESTER

COURSE 1 : ESSENTIALS AND APPLICATIONS OF MATHEMATICAL, PHYSICAL AND CHEMICAL SCIENCES

Hours: 5hrs/week Credits: 4

Course Objective:

The objective of this course is to provide students with a comprehensive understanding of the essential concepts and applications of mathematical, physical, and chemical sciences. The course aims to develop students' critical thinking, problem-solving, and analytical skills in these areas, enabling them to apply scientific principles to real-world situations.

Learning outcomes:

1. Apply critical thinking skills to solve complex problems involving complex numbers, trigonometric ratios, vectors, and statistical measures.
2. To Explain the basic principles and concepts underlying a broad range of fundamental areas of physics and to Connect their knowledge of physics to everyday situations
3. To Explain the basic principles and concepts underlying a broad range of fundamental areas of chemistry and to Connect their knowledge of chemistry to daily life.
4. Understand the interplay and connections between mathematics, physics, and chemistry in various applications. Recognize how mathematical models and physical and chemical
5. principles can be used to explain and predict phenomena in different contexts.
6. To explore the history and evolution of the Internet and to gain an understanding of network security concepts, including threats, vulnerabilities, and countermeasures.

UNIT I: ESSENTIALS OF MATHEMATICS:9hrs

Complex Numbers: Introduction of the new symbol i – General form of a complex number – Modulus- Amplitude form and conversions

Trigonometric Ratios: Trigonometric Ratios and their relations – Problems on calculation of angles Vectors: Definition of vector addition – Cartesian form – Scalar and vector product and problems Statistical Measures: Mean, Median, Mode of a data and problems

UNIT II: ESSENTIALS OF PHYSICS: 9hrs

Definition and Scope of Physics- Measurements and Units - Motion of objects: Newtonian Mechanics and relativistic mechanics perspective - Laws of Thermodynamics and Significance-Acoustic waves and electromagnetic waves- Electric and Magnetic fields and their interactions-Behaviour of atomic and nuclear particles- Wave-particle duality, the uncertainty principle-Theories and understanding of universe

UNIT III: ESSENTIALS OF CHEMISTRY: : 9hrs

Definition and Scope of Chemistry- Importance of Chemistry in daily life -Branches of chemistry and significance- Periodic Table- Electronic Configuration, chemical changes, classification of matter, Biomolecules- carbohydrates, proteins, fats and vitamins.

UNIT IV: APPLICATIONS OF MATHEMATICS, PHYSICS & CHEMISTRY: 9hrs

Applications of Mathematics in Physics & Chemistry: Calculus , Differential Equations & Complex Analysis

Application of Physics in Industry and Technology: Electronics and Semiconductor Industry, Robotics and Automation, Automotive and Aerospace Industries, Quality Control and Instrumentation, Environmental Monitoring and Sustainable Technologies.

Application of Chemistry in Industry and Technology: Chemical Manufacturing, Pharmaceuticals and Drug Discovery, Materials Science, Food and Beverage Industry.

UNIT V: ESSENTIALS OF COMPUTER SCIENCE:

Milestones of computer evolution - Internet, history, Internet Service Providers, Types of Networks, IP, Domain Name Services, applications.

Ethical and social implications: Network and security concepts- Information Assurance Fundamentals, Cryptography-Symmetric and Asymmetric, Malware, Firewalls, Fraud Techniques- Privacy and Data Protection

Recommended books:

1. Functions of one complex variable by John.B.Conway, Springer- Verlag.
2. Elementary Trigonometry by H.S.Hall and S.R.Knight
3. Vector Algebra by A.R.Vasishtha, Krishna Prakashan Media(P)Ltd. 4.Basic Statistics by B.L.Agarwal, New age international Publishers
4. University Physics with Modern Physics by Hugh D. Young and Roger A. Freedman
5. Fundamentals of Physics by David Halliday, Robert Resnick, and Jearl Walker
6. Physics for Scientists and Engineers with Modern Physics" by Raymond A. Serway and John W. Jewett Jr.
7. Physics for Technology and Engineering" by John Bird
8. Chemistry in daily life by Kirpal Singh
9. Chemistry of bio molecules by S. P. Bhutan
10. Fundamentals of Computers by V. Raja Raman
11. Cyber Security Essentials by James Graham, Richard Howard, Ryan Olson

STUDENT ACTIVITIES
UNIT I: ESSENTIALS OF MATHEMATICS:

1: Complex Number Exploration Provide students with a set of complex numbers in both rectangular and polar forms.They will plot the complex numbers on the complex plane and identify their properties

2: Trigonometric Ratios Problem Solving Give students a set of problems that require the calculation of trigonometric ratios and their relations.

Students will solve the problems using the appropriate trigonometric functions (sine, cosine, tangent, etc.) and trigonometric identities.

3: Vector Operations and Applications Provide students with a set of vectors in Cartesian form.

Students will perform vector addition and subtraction operations to find the resultant vectors. They will also calculate the scalar and vector products of given vectors.

4: Statistical Measures and Data Analysis Give students a dataset containing numerical values.

Students will calculate the mean, median, and mode of the data, as well as other statistical measures if appropriate (e.g., range, standard deviation).

They will interpret the results and analyze the central tendencies and distribution of the data.

UNIT II: ESSENTIALS OF PHYSICS:

Concept Mapping

Divide students into groups and assign each group one of the topics.

Students will create a concept map illustrating the key concepts, relationships, and applications related to their assigned topic.

Encourage students to use visual elements, arrows, and labels to represent connections and interdependencies between concepts.

Laboratory Experiment

Select a laboratory experiment related to one of the topics, such as motion of objects or electric and magnetic fields.

Provide the necessary materials, instructions, and safety guidelines for conducting the experiment.

Students will work in small groups to carry out the experiment, collect data, and analyze the results.

After the experiment, students will write a lab report summarizing their findings, observations, and conclusions.

UNIT III: ESSENTIALS OF CHEMISTRY

1: Chemistry in Daily Life Presentation

Divide students into groups and assign each group a specific aspect of daily life where chemistry plays a significant role, such as food and nutrition, household products, medicine, or environmental issues.

Students will research and create a presentation (e.g., PowerPoint, poster, or video) that showcases the importance of chemistry in their assigned aspect.

2: Periodic Table Exploration

Provide students with a copy of the periodic table.

Students will explore the periodic table and its significance in organizing elements based on their properties.

They will identify and analyze trends in atomic structure, such as electronic configuration, atomic size, and ionization energy.

3: Chemical Changes and Classification of Matter

Provide students with various substances and chemical reactions, such as mixing acids and bases or observing a combustion reaction.

Students will observe and describe the chemical changes that occur, including changes in color, temperature, or the formation of new substances.

4: Biomolecules Investigation

Assign each student or group a specific biomolecule category, such as carbohydrates, proteins, fats, or vitamins.

Students will research and gather information about their assigned biomolecule category, including its structure, functions, sources, and importance in the human body.

They can create informative posters or presentations to present their findings to the class.

UNIT IV: APPLICATIONS OF MATHEMATICS, PHYSICS & CHEMISTRY

1: Interdisciplinary Case Studies

Divide students into small groups and provide them with interdisciplinary case studies that involve the interdisciplinary application of mathematics, physics, and chemistry.

Each case study should present a real-world problem or scenario that requires the integration of concepts from all three disciplines.

2: Design and Innovation Project

Challenge students to design and develop a practical solution or innovation that integrates mathematics, physics, and chemistry principles.

Students can choose a specific problem or area of interest, such as renewable energy, environmental conservation, or materials science.

3: Laboratory Experiments

Assign students laboratory experiments that demonstrate the practical applications of mathematics, physics, and chemistry.

Examples include investigating the relationship between concentration and reaction rate, analyzing the behavior of electrical circuits, or measuring the properties of materials.

.4: Mathematical Modeling

Present students with real-world problems that require mathematical modeling and analysis.

UNIT V: ESSENTIALS OF COMPUTER SCIENCE:

Identifying the attributes of network (Topology, service provider, IP address and bandwidth of your college network) and prepare a report covering network architecture.

Identify the types of malwares and required firewalls to provide security.

Latest Fraud techniques used by hackers.

I Semester

Course 2: ADVANCES IN MATHEMATICAL, PHYSICAL AND CHEMICAL SCIENCES

Hours: 5 hrs/week Credits: 4

Course Objective:

The objective of this course is to provide students with an in-depth understanding of the recent advances and cutting-edge research in mathematical, physical, and chemical sciences. The course aims to broaden students' knowledge beyond the foundational concepts and expose them to the latest developments in these disciplines, fostering critical thinking, research skills, and the ability to contribute to scientific advancements.

Learning outcomes:

Explore the applications of mathematics in various fields of physics and chemistry, to understand how mathematical concepts are used to model and solve real-world problems.

To Explain the basic principles and concepts underlying a broad range of fundamental areas of physics and to Connect their knowledge of physics to everyday situations.

Understand the different sources of renewable energy and their generation processes and advances in nanomaterials and their properties, with a focus on quantum dots. To study the emerging field of quantum communication and its potential applications. To gain an understanding of the principles of biophysics in studying biological systems. Explore the properties and applications of shape memory materials.

Understand the principles and techniques used in computer-aided drug design and drug delivery systems, to understand the fabrication techniques and working principles of nanosensors. Explore the effects of chemical pollutants on ecosystems and human health.

Understand the interplay and connections between mathematics, physics, and chemistry in various advanced applications. Recognize how mathematical models and physical and chemical principles can be used to explain and predict phenomena in different contexts.

5 Understand and convert between different number systems, such as binary, octal, decimal, and hexadecimal. Differentiate between analog and digital signals and understand their characteristics.Gain knowledge of different types of transmission media, such as wired (e.g., copper cables, fiber optics) and wireless (e.g., radio waves, microwave, satellite)..

UNIT I: ADVANCES IN BASICS MATHEMATICS 9hrs

Straight Lines: Different forms – Reduction of general equation into various forms –Point of intersection of two straight lines

Limits and Differentiation: Standard limits – Derivative of a function –Problems on product rule and quotient rule

Integration: Integration as a reverse process of differentiation – Basic methods of integration

Matrices: Types of matrices – Scalar multiple of a matrix – Multiplication of matrices – Transpose of a matrix and determinants

UNIT II: ADVANCES IN PHYSICS: 9hrs

Renewable energy: Generation, energy storage, and energy-efficient materials and devices. Recent advances in the field of nanotechnology: Quantum dots, Quantum Communication- recent advances in biophysics- recent advances in medical physics- Shape Memory Materials.

UNIT III: ADVANCES IN CHEMISTRY: 9hrs

Computer aided drug design and delivery, nano sensors, Chemical Biology, impact of chemical pollutants on ecosystems and human health, Dye removal - Catalysis method

UNIT IV: ADVANCED APPLICATIONS OF MATHEMATICS, PHYSICS & CHEMISTRY 9hrs

Mathematical Modelling applications in physics and chemistry Application of Renewable energy: Grid Integration and Smart Grids, Application of nanotechnology: Nanomedicine,

Application of biophysics: Biophysical Imaging, Biomechanics, Neurophysics,

Application of medical physics: Radiation Therapy, Nuclear medicine

Solid waste management, Environmental remediation- Green Technology, Water treatment.

UNIT V: Advanced Applications of computer Science 9hrs

Number System-Binary, Octal, decimal, and Hexadecimal, Signals-Analog, Digital, Modem, Codec, Multiplexing, Transmission media, error detection and correction- Parity check and CRC, Networking devices-Repeater, hub, bridge, switch, router, gateway.

Recommended books:

1. Coordinate Geometry by S.L.Lony, Arihant Publications
2. Calculus by Thomas and Finny, Pearson Publications
3. Matrices by A.R.Vasishtha and A.K.Vasishtha, Krishna Prakashan Media(P)Ltd.
4. "Renewable Energy: Power for a Sustainable Future" by Godfrey Boyle
5. "Energy Storage: A Nontechnical Guide" by Richard Baxter
6. "Nanotechnology: Principles and Applications" by Sulabha K. Kulkarni and Raghvendra A. Bohara
7. "Biophysics: An Introduction" by Rodney Cotterill

8. "Medical Physics: Imaging" by James G. Webster
9. "Shape Memory Alloys: Properties and Applications" by Dimitris C. Lagoudas
10. Nano materials and applications by M.N.Borah
11. Environmental Chemistry by Anil.K.D.E.
12. Digital Logic Design by Morris Mano
13. Data Communication & Networking by Bahrouz Forouzan.

STUDENT ACTIVITIES

UNIT I: ADVANCES IN BASIC MATHEMATICS

1: Straight Lines Exploration

Provide students with a set of equations representing straight lines in different forms, such as slope-intercept form, point-slope form, or general form.

Students will explore the properties and characteristics of straight lines, including their slopes, intercepts, and point of intersection.

2: Limits and Differentiation Problem Solving

Students will apply the concept of limits to solve various problems using standard limits.

Encourage students to interpret the results and make connections to real-world applications, such as analyzing rates of change or optimizing functions.

3: Integration Exploration

Students will explore the concept of integration as a reverse process of differentiation and apply basic methods of integration, such as the product rule, substitution method, or
integration by parts.

Students can discuss the significance of integration in various fields, such as physics and chemistry

4: Matrices Manipulation

Students will perform operations on matrices, including scalar multiplication, matrix multiplication, and matrix transpose.

Students can apply their knowledge of matrices to real-world applications, such as solving systems of equations or representing transformations in geometry.

UNIT II: ADVANCES IN PHYSICS:

1: Case Studies

Provide students with real-world case studies related to renewable energy, nanotechnology, biophysics, medical physics, or shape memory materials.

Students will analyze the case studies, identify the challenges or problems presented, and propose innovative solutions based on the recent advances in the respective field.

They will consider factors such as energy generation, energy storage, efficiency,
sustainability, materials design, biomedical applications, or technological advancements.

2: Experimental Design

Assign students to design and conduct experiments related to one of the topics: renewable energy, nanotechnology, biophysics, medical physics, or shape memory materials.

They will identify a specific research question or problem to investigate and design an experiment accordingly.

Students will collect and analyze data, interpret the results, and draw conclusions based on their findings.

They will discuss the implications of their experimental results in the context of recent advances in the field.

3: Group Discussion and Debate

Organize a group discussion or debate session where students will discuss the ethical, social, and environmental implications of the recent advances in renewable energy, nanotechnology, biophysics, medical physics, and shape memory materials.

Assign students specific roles, such as proponent, opponent, or moderator, and provide them with key points and arguments to support their positions.

UNIT III: ADVANCES IN CHEMISTRY:

Experimental Design and Simulation

In small groups, students will design experiments or simulations related to the assigned topic.

For example, in the context of computer-aided drug design, students could design a virtual screening experiment to identify potential drug candidates for a specific disease target.

For nano sensors, students could design an experiment to demonstrate the sensitivity and selectivity of nano sensors in detecting specific analytes.

Chemical biology-related activities could involve designing experiments to study enzyme- substrate interactions or molecular interactions in biological systems.

Students will perform their experiments or simulations, collect data, analyze the results, and draw conclusions based on their findings.

Case Studies and Discussion

Provide students with real-world case studies related to the impact of chemical pollutants on ecosystems and human health.

Students will analyze the case studies, identify the sources and effects of chemical pollutants, and propose mitigation strategies to minimize their impact.

Encourage discussions on the ethical and environmental considerations when dealing with chemical pollutants.

For the dye removal using the catalysis method, students can explore case studies where catalytic processes are used to degrade or remove dyes from wastewater.

Students will discuss the principles of catalysis, the advantages and limitations of the catalysis method, and its applications in environmental remediation.

3: Group Project

Assign students to work in groups to develop a project related to one of the topics.

The project could involve designing a computer-aided drug delivery system, developing a nano sensor for a specific application, or proposing strategies to mitigate the impact of chemical pollutants on ecosystems.

Students will develop a detailed project plan, conduct experiments or simulations, analyze data, and present their findings and recommendations.

Encourage creativity, critical thinking, and collaboration throughout the project.

UNIT IV: ADVANCED APPLICATIONS OF MATHEMATICS, PHYSICS & CHEMISTRY

1: Mathematical Modelling Experiment

Provide students with a mathematical modelling experiment related to one of the topics. For example, in the context of renewable energy, students can develop a mathematical model to optimize the placement and configuration of solar panels in a solar farm.

Students will work in teams to design and conduct the experiment, collect data, and analyze the results using mathematical models and statistical techniques.

They will discuss the accuracy and limitations of their model, propose improvements, and
interpret the implications of their findings in the context of renewable energy or the specific application area.

2: Case Studies and Group Discussions

Assign students to analyze case studies related to the applications of mathematical modelling in nanotechnology, biophysics, medical physics, solid waste management, environmental remediation, or water treatment.

Students will discuss the mathematical models and computational methods used in the case studies, analyze the outcomes, and evaluate the effectiveness of the modelling approach.

Encourage group discussions on the challenges, ethical considerations, and potential advancements in the field.

Students will present their findings and engage in critical discussions on the advantages and limitations of mathematical modelling in solving complex problems in these areas.

Group Project

Assign students to work in groups to develop a group project that integrates mathematical modelling with one of the application areas: renewable energy, nanotechnology, biophysics, medical physics, solid waste management, environmental remediation, or water treatment.

The project could involve developing a mathematical model to optimize the delivery of radiation therapy in medical physics or designing a mathematical model to optimize waste management practices.

Students will plan and execute their project, apply mathematical modelling techniques, analyze the results, and present their findings and recommendations.

Encourage creativity, critical thinking, and collaboration throughout the project.

UNIT V: Advanced Applications of computer Science

Students must be able to convert numbers from other number system to binary number systems

Identify the networking media used for your college network

Identify all the networking devices used in your college premises.

II Semester

Course 3: Problem Solving using C

Credits -3

Course Objectives

1. To explore basic knowledge on computers
2. Learn how to solve common types of computing problems.
3. Learn to map problems to programming features of C.
4. Learn to write good portable C programs.

Course Outcomes

Upon successful completion of the course, a student will be able to:

1. Understand the working of a digital computer and Fundamental constructs of Programming
2. Analyze and develop a solution to a given problem with suitable control structures
3. Apply the derived data types in program solutions
4. Use the 'C' language constructs in the right way
5. Apply the Dynamic Memory Management for effective memory utilization

UNIT-I

Introduction to computer and programming: Introduction, Basic block diagram and functions of various components of computer, Concepts of Hardware and software, Types of software, Compiler and interpreter, Concepts of Machine level, Assembly level and high-level programming, Flowcharts and Algorithms

Fundamentals of C: History of C, Features of C, C Tokens-variables and keywords and identifiers, constants and Data types, Rules for constructing variable names, Operators, Structure of C program, Input /output statements in C-Formatted and Unformatted I/O

UNIT-II

Control statements: Decision making statements: if, if else, else if ladder, switch statements. Loop control statements: while loop, for loop and do-while loop. Jump Control statements: break, continue and goto.

UNIT-III

Derived data types in C: Arrays: One Dimensional arrays - Declaration, Initialization and Memory representation; Two Dimensional arrays -Declaration, Initialization and Memory representation.

Strings: Declaring & Initializing string variables; String handling functions, Character handling functions

UNIT-IV

Functions: Function Prototype, definition and calling. Return statement. Nesting of functions. Categories of functions. Recursion, Parameter Passing by address & by value. Local and Global variables. **Storage classes**: automatic, external, static and register.

Pointers: Pointer data type, Pointer declaration, initialization, accessing values using pointers. Pointer arithmetic. Pointers and arrays, pointers and functions.

UNIT-V

Dynamic Memory Management: Introduction, Functions-malloc, calloc, realloc, free **Structures:** Basics of structure, structure members, accessing structure members, nested structures, array of structures, structure and functions, structures and pointers. **Unions** - Union definition; difference between Structures and Unions.

Text Books:

1. E. Balagurusamy, “Programming in ANSI C”, Tata McGraw Hill, 6th Edn, ISBN-13: 978-1-25- 90046-2
2. Herbert Schildt, ―Complete Reference with C, Tata McGraw Hill, 4th Edn., ISBN- 13: 9780070411838, 2000
3. Computer fundamentals and programming in C, REEMA THAREJA, OXFORD UNIVERSITY PRESS

Reference Books

1. E Balagurusamy, COMPUTING FUNDAMENTALS & C PROGRAMMING – Tata McGraw-Hill, Second Reprint 2008, ISBN 978-0-07-066909-3.
2. Ashok N Kamthane, Programming with ANSI and Turbo C, Pearson Edition Publ, 2002.
3. Henry Mullish&Huubert L.Cooper: The Spirit of C An Introduction to modern Programming, Jaico Pub. House,1996.
4. Y kanithkar, let us C BPB, 13 th edition-2013, ISBN:978-8183331630,656 pages.

SUGGESTED CO-CURRICULAR ACTIVITIES & EVALUATION METHODS:

Unit 1: Activity: Quiz on computer hardware and software concepts

Evaluation Method: Objective-based quiz assessing knowledge and understanding

Unit 2: Activity: Problem-solving using Decision-Making Statements

Evaluation Method: Correctness of decision-making logic

Unit 3: Activity: Array and String Program Debugging

Evaluation Method: Identification and correction of errors in code

Unit 4: Activity: Pair Programming Exercise on Functions

Evaluation Method: Collaboration and Code Quality

Unit 5: Activity: Structured Programming Assignment

Evaluation Method: Appropriate use of structures and nested structures

II Semester

Course 3: Problem Solving using C

Credits -1

List of Experiments

1. A. Write a program to calculate simple & compound interest
   B. Write a C program to interchange two numbers.
2. Find the biggest of three numbers using C.
3. Write a c program to find the sum of individual digits of a positive integer.
4. A Fibonacci sequence is defined as follows: the first and second terms in the sequenceare 0 and 1. Subsequent terms are found by adding the preceding two terms in the sequence.
5. Write a c program to check whether a number is Armstrong or not.
6. Write a c program to generate all the prime numbers between 1 and n, where n is avalue supplied by the user.
7. Write a c program that implements searching of given item in given list
8. Write a c program that uses functions to perform the following: Addition of twomatrices. Multiplication of two matrices.
9. Write a program for concatenation of two strings.
10. Write a program for length of a string with and without String Handling functions
11. Write a program to demonstrate Call by Value and Call by Reference mechanism
12. Write a Program to find GCD of Two numbers using Recursion
13. Write a c program to perform various operations using pointers.
14. Write a c program to read data of 10 employees with a structure of 1.employee id2.aadar no, 3.title, 4.joined date, 5.salary, 6.date of birth, 7.gender, 8.department.
15. Write a Program to demonstrate dynamic arrays using Dynamic Memory Management functions

II Semester
Course 4: Digital Logic Design
Credits -3

Course Objectives

To familiarize with the concepts of designing digital circuits.

Course Outcomes

Upon successful completion of the course, the students will be able to

1. Understand how to Convert numbers from one radix to another radix and performarithmetic operations.
2. Simplify Boolean functions using Boolean algebra and k- maps
3. Design adders and subtractors circuits
4. Design combinational logic circuits such as decoders, encoders, multiplexers and demultiplexers.
5. Use flip flops to design registers and counters.

UNIT – I

Number Systems: Binary, octal, decimal, hexadecimal number systems, conversion of numbers from one radix to another radix, r's, (r-1)'s complements, signed binary numbers, addition and subtraction of unsigned and signed numbers, weighted and unweighted codes.

UNIT – II

Logic Gates and Boolean Algebra: NOT, AND, OR, universal gates, X-OR and X-NOR gates, Boolean laws and theorems, complement and dual of a logic function, canonical and standard forms, two level realization of logic functions using universal gates, minimizations of logic functions (POS and SOP) using Boolean theorems, K-map (up to four variables), don't care conditions.

UNIT – III

Combinational Logic Circuits – 1: Design of half adder, full adder, half subtractor, fullsubtractor, ripple adders and subtractors, ripple adder / subtractor.

UNIT – IV

Combinational Logic Circuits – 2: Design of decoders, encoders, priority encoder, multiplexers, demultiplexers, higher order decoders, demultiplexers and multiplexers, realization of Boolean functions using decoders, multiplexers.

UNIT – V

Sequential Logic Circuits: Classification of sequential circuits, latch and flip-flop, RS- latch using NAND and NOR Gates, truth tables, RS, JK, T and D flip-flops, truth and excitation tables, conversion of flip- flops, flip-flops with asynchronous inputs (preset and clear).

Design of registers, shift registers, bidirectional shift registers, universal shift register, design of ripple counters, synchronous counters and variable modulus counters.

Text Books:

1. M. Morris Mano, Michael D Ciletti, “Digital Design”, 5th edition, PEA.

Reference Books

1. Kohavi, Jha, “Switching and Finite Automata Theory”, 3rd edition, Cambridge.
2. 2. Leach, Malvino, Saha, “Digital Principles and Applications”, 7th edition, TMH.
3. 3. Roth, “Fundamentals of Logic Design”, 5th edition, Cengage.

SUGGESTED CO-CURRICULAR ACTIVITIES & EVALUATION METHODS:

Unit 1: Activity: JAM (Just a Minute) Session: Explaining Radix Conversion

Evaluation Method: Communication Skills and Knowledge Presentation

Unit 2: Activity: Boolean Algebra Assignment

Evaluation Method: Assignment Completion and Correctness

Unit 3: Activity: Hands-on Lab Activity: Building Adder and Subtractor Circuits

Evaluation Method: Lab Performance and Correctness of Circuit Implementation

Unit 4: Activity: Group Discussion: Applications of Decoders, Encoders, Multiplexers

Evaluation Method: Participation and Critical Thinking

Unit 5: Activity: Quiz on Flip-Flops and Register-Counter Design

Evaluation Method: Quiz Performance and Knowledge Retention

II Semester

Course 4: Digital Logic Design

Credits -1

List of Experiments

The laboratory work can be done by using physical gates and necessary equipment or simulators.

Simulators: https://sourceforge.net/projects/gatesim/ or https://circuitverse.org/ or any free open-source simulator

1. Introduction to digital electronics lab- nomenclature of digital ICs, specifications, study of the data sheet, concept of Vcc and ground, verification of the truth tables of logic gates using TTL ICs.
2. Implementation of the given Boolean functions using logic gates in both SOP and POS forms
3. Realization of basic gates using universal gates.
4. Design and implementation of half and full adder circuits using logic gates.
5. Design and implementation of half and full subtractor circuits using logic gates.
6. Verification of stable tables of RS, JK, T and D flip-flops using NAND gates.
7. Verification of stable tables of RS, JK, T and D flip-flops using NOR gates.

8. Implementation and verification of Decoder and encoder using logic gates.
9. Implementation of 4X1 MUX and DeMUX using logic gates.
10. Implementation of 8X1 MUX using suitable lower order MUX.
11. Implementation of 7-segment decoder circuit.
12. Implementation of 4-bit parallel adder.
13. Design and verification of 4-bit synchronous counter.
14. Design and verification of 4-bit asynchronous counter.

III Semester
Course 5: Object Oriented Programming using Java

Credits -3

Course Objectives

To introduce the fundamental concepts of Object-Oriented programming and to design &implement object-oriented programming concepts in Java.

Course Outcomes

Upon successful completion of the course, a student will be able to:

1. Understand the basic concepts of Object-Oriented Programming and Java Program Constructs
2. Implement classes and objects and analyze Inheritance and Dynamic Method Dispatch
3. Demonstrate various classes in different packages and can design own packages
4. Manage Exceptions and Apply Threads
5. Create GUI screens along with event handling

UNIT-I

OOPs Concepts and Java Programming: Introduction to Object-Oriented concepts, proceduraland object-oriented programming paradigm

Java programming: An Overview of Java, Java Environment, Data types, Variables, constants, scope and life time of variables, operators, type conversion and casting, Accepting Input from the Keyboard, Reading Input with Java.util.Scanner Class, Displaying Output with System.out.printf(), Displaying Formatted Output with String.format(), Control Statements

UNIT-II

Arrays, Command Line Arguments, Strings-String Class Methods

Classes & Objects: Creating Classes, declaring objects, Methods, parameter passing, static fieldsand methods, Constructors, and 'this' keyword, overloading methods and access

Inheritance: Inheritance hierarchies, super and subclasses, member access rules, 'super' keyword, preventing inheritance: final classes and methods, the object class and its methods; **Polymorphism**: Dynamic binding, method overriding, abstract classes and methods;

UNIT-III

Interface: Interfaces VS Abstract classes, defining an interface, implement interfaces, accessing implementations through interface references, extending interface;

Packages: Defining, creating and accessing a package, understanding CLASSPATH, importing packages.

Exception Handling: Benefits of exception handling, the classification of exceptions, exception hierarchy, checked exceptions and unchecked exceptions, usage of try, catch, throw, throws and finally, rethrowing exceptions, exception specification, built in exceptions, creating own exceptionsub classes.

UNIT-IV

Multithreading: Differences between multiple processes and multiple threads, thread states, thread life cycle, creating threads, interrupting threads, thread priorities, synchronizing threads, inter thread communication.

Stream based I/O (java.io) – The Stream classes-Byte streams and Character streams, Reading console Input and Writing Console Output, File class, Reading and writing Files, The Console class, Serialization

UNIT-V

GUI Programming with Swing- Introduction, MVC architecture, components, containers. Understanding Layout Managers - Flow Layout, Border Layout, Grid Layout, Card Layout, GridBag Layout.

Event Handling- The Delegation event model- Events, Event sources, Event Listeners, Event classes, Handling mouse and keyboard events, Adapter classes, Inner classes, Anonymous Inner classes.

Text Books:

1. Java The complete reference, 9th edition, Herbert Schildt, McGraw Hill.
2. Understanding Object-Oriented Programming with Java, updated edition, T. Budd, Pearson Education.

Reference Books

1. Cay S. Horstmann, “Core Java Fundamentals”, Volume 1, 11 th Edition, Prentice Hall, 2018.
2. Paul Deitel, Harvey Deitel, “Java SE 8 for programmers”, 3rd Edition, Pearson, 2015.
3. S. Malhotra, S. Chudhary, Programming in Java, 2nd edition, Oxford Univ. Press.

SUGGESTED CO-CURRICULAR ACTIVITIES & EVALUATION METHODS:

Unit 1: Activity: Quiz on Object-Oriented Programming Concepts and Java Constructs

Evaluation Method: Quiz Performance and Knowledge Retention

Unit 2: Activity: Object-Oriented Programming Assignment: Class Implementation

Evaluation Method: Assignment Completion and Correctness

Unit 3: Activity: Hands-on Lab Activity: Creating and Using Custom Java Packages

Evaluation Method: Lab Performance and Correctness of Code Implementation

Unit 4: Activity: Case Study Discussion on where multi-threading is crucial

Evaluation Method: Critical thinking, problem-solving, and presentation skills.

Unit 5: Activity: GUI design contest using Java Swings

Evaluation Method: GUI design, Visual appearance and user friendliness, usability, and adherence to event handling principles.

III Semester
Course 5: Object Oriented Programming using Java Lab

Credits -1

List of Experiments

1. Write a Java program to print Fibonacci series using for loop.
2. Write a Java program to calculate multiplication of 2 matrices.
3. Create a class Rectangle. The class has attributes length and width. It should have methods that calculate the perimeter and area of the rectangle. It should have read Attributes method to read length and width from user.
4. Write a Java program that implements method overloading.
5. Write a Java program for sorting a given list of names in ascending order.
6. Write a Java program that displays the number of characters, lines and words in a text file.
7. Write a Java program to implement various types of inheritance

   i. Single   ii. Multi-Level   iii. Hierarchical   iv. Hybrid
8. Write a java program to implement runtime polymorphism.
9. Write a Java program which accepts withdraw amount from the user and throws an exception "In Sufficient Funds" when withdraw amount more than available amount.
10. Write a Java program to create three threads and that displays "good morning", for every one second, "hello" for every 2 seconds and "welcome" for every 3 seconds by using extending Thread class.
11. Write a Java program that creates three threads. First thread displays "OOPS", the second thread displays "Through" and the third thread Displays "JAVA" by using Runnable interface.
12. Implement a Java program for handling mouse events when the mouse entered, exited, clicked, pressed, released, dragged and moved in the client area.
13. Implement a Java program for handling key events when the key board is pressed, released, typed.
14. Write a Java swing program that reads two numbers from two separate text fields and display sum of two numbers in third text field when button "add" is pressed.
15. Write a Java program to design student registration form using Swing Controls. The form which having the following fields and button SAVE

    Form Fields are: Name, RNO, Mailid, Gender, Branch, Address.

III Semester
Course 6: Data Structures using C
Credits -3

Course Objectives

To introduce the fundamental concept of data structures and to emphasize the importance of various data structures in developing and implementing efficient algorithms.

Course Outcomes

Upon successful completion of the course, a student will be able to:

1. Understand various Data Structures for data storage and processing.
2. Realize Linked List Data Structure for various operations
3. Analyze step by step and develop algorithms to solve real world problems by implementing Stacks, Queues data structures.
4. Understand and implement various searching & sorting techniques.
5. Understand the Non-Linear Data Structures such as Binary Trees and Graphs

UNIT-I

Basic Concepts: Pointers and dynamic memory allocation, Algorithm-Definition and characteristics, Algorithm Analysis-Space Complexity, Time Complexity, Asymptotic Notation **Introduction to Data structures:** Definition**,** Types of Data structure, Abstract Data Types (ADT), Difference between Abstract Data Types, Data Types, and Data Structures.

Arrays-Concept of Arrays, Single dimensional array, Two dimensional array, Operations on arrays with Algorithms (searching, traversing, inserting, deleting)

UNIT-II

Linked List: Concept of Linked Lists, Representation of linked lists in Memory, Comparison between Linked List and Array, Types of Linked Lists - Singly Linked list, Doubly Linked list, Circularly Singly Linked list, Circularly Doubly Linked list;

Implementation of Linked List ADT: Creating a List, Traversing a linked list, Searching linkedlist, Insertion and deletion into linked list (At first Node, Specified Position, Last node), Application of linked lists

UNIT-III

Stacks: Introduction to stack ADT, Representation of stacks with array and Linked List, Implementation of stacks, Application of stacks - Polish Notations - Converting Infix to Post Fix Notation - Evaluation of Post Fix Notation - Tower of Hanoi, Recursion: Concept and Comparison between recursion and Iteration

Queues: Introduction to Queue ADT, Representation of Queues with array and Linked List, Implementation of Queues, Application of Queues Types of Queues- Circular Queues, De-queues, Priority Queue

UNIT-IV

Searching: Linear or Sequential Search, Binary Search and Indexed Sequential Search

Sorting: Selection Sort, Bubble Sort, Insertion Sort, Quick Sort and Merge Sort

UNIT-V

Binary Trees: Concept of Non- Linear Data Structures, Introduction Binary Trees, Types of Trees, Basic Definition of Binary Trees, Properties of Binary Trees, Representation of Binary Trees, Operations on a Binary Search Tree, Binary Tree Traversal, Applications of Binary Tree. **Graphs:** Introduction to Graphs, Terms Associated with Graphs, Sequential Representation of Graphs, Linked Representation of Graphs, Traversal of Graphs (DFS, BFS), Application of Graphs.

Text Books:

1. Horowitz and Sahani, “Fundamentals of Data Structures”, Galgotia Publications Pvt Ltd Delhi India.
2. A.K. Sharma ,Data Structure Using C, Pearson Education India.
3. “Data Structures Using C” Balagurusamy E. TMH

Reference Books

1. “Data Structures through C”, Yashavant Kanetkar, BPB Publications
2. Rajesh K. Shukla, “Data Structure Using C and C++” Wiley Dreamtech Publication.
3. Lipschutz, “Data Structures” Schaum’s Outline Series, Tata Mcgraw-hill Education (India)Pvt. Ltd .
4. Michael T. Goodrich, Roberto Tamassia, David M. Mount “Data Structures and Algorithms in C++”, Wiley India.

SUGGESTED CO-CURRICULAR ACTIVITIES & EVALUATION METHODS:

Unit 1: Activity: Algorithm analysis exercises

Evaluation Method: Programming Assignment and Correctness

Unit 2: Activity: Presentations on real-life applications of linked lists

Evaluation Method: Presentation skills or reports

Unit 3: Activity: Role-playing activities for stack operations

Evaluation Method: Problem-solving skills, communication and collaboration abilities.

Unit 4: Activity: Sorting algorithm analysis and comparison activities

Evaluation Method: Performance analysis and presentation.

Unit 5: Activity: Case Study on Applications of Graphs

Evaluation Method: Critical thinking, problem-solving, and presentation skills

III Semester
Course 6: Data Structures Using C

Credits -1

List of Experiments:

1. Write a program to read 'N' numbers of elements into an array and also perform the following operation on an array
   a. Add an element at the beginning of an array
   b. Insert an element at given index of array
   c. Update an element using a values and index
   d. Delete an existing element
2. Write Program to implement Single Linked List with insertion, deletion and traversal operations
3. Write Program to implement Circular doubly Linked List with insertion, deletion and traversal operations
4. Write Programs to implement the Stack operations using an array
5. Write a program using stacks to convert a given infix expression to postfix
6. Write Programs to implement the Stack operations using Liked List.
7. Write Programs to implement the Queue operations using an array.
8. Write Programs to implement the Queue operations using Liked List.
9. Write a program for Binary Search Tree Traversals
10. Write a program to search an item in a given list using the following Searching Algorithms
    a. Linear Search
    b. Binary Search.
11. Write a program for implementation of the following Sorting Algorithms
    a. Bubble Sort
    b. Insertion Sort
    c. Quick Sort

III Semester
Course 7: Computer Organization
Credits -3

Course Objectives

To familiarize with organizational aspects of memory, processor and I/O.

Course Outcomes

Upon successful completion of the course, the students will be able to

1. Identify different types of instructions
2. Differentiate between micro-programmed and hard-wired control units.
3. Analyse the performance of hierarchical organization of memory.
4. Summarize different data transfer techniques.
5. Demonstrate arithmetic operations on fixed- and floating-point numbers and illustrate concepts of parallel processing.

UNIT – I

Register Transfer Language and Micro Operations: Introduction- Functional units, computer registers, register transfer language, register transfer, bus and memory transfers, arithmetic, logic and shift micro-operations, arithmetic logic shift unit.

Basic Computer Organization and Design: Instruction codes, instruction cycle.

Register reference instructions, Memory – reference instructions, input – output and interrupt.

UNIT – II

CPU and Micro Programmed Control: Central Processing unit: Introduction, instruction formats, addressing modes.Control memory, address sequencing, design of control unit - hard wired control, micro programmed control.

UNIT – III

Memory Organization: Memory hierarchy, main memory, auxiliary memory, associative memory, cache Memory and mappings.

UNIT – IV

Input-Output Organization: Peripheral Devices, input-output interface, asynchronous datatransfer, modes of transfer- programmed I/O, priority interrupt, direct memory access, Input – Output Processor (IOP).

UNIT – V

Computer Arithmetic and Parallel Processing: Data representation- fixed point, floating point, addition and subtraction, multiplication and division algorithms.

Parallel Processing-Parallel Processing, Pipelining, Arithmetic Pipeline, Instruction Pipeline.

Text Books:

1. M. Moris Mano, “Computer Systems Architecture”, 3rd edition, Pearson/ PHI.

Reference Books:

1. Carl Hamacher, ZvonksVranesic, SafeaZaky, “Computer Organization”, 5th edition,McGraw Hill.
2. William Stallings, “Computer Organization and Architecture”, 8th edition, Pearson/PHI.

SUGGESTED CO-CURRICULAR ACTIVITIES & EVALUATION METHODS:

Unit 1: Activity: Quiz competition on micro-operations.

Evaluation Method: Accuracy and speed in answering quiz questions.

Unit 2: Activity: Instruction Format Puzzle: Solving a puzzle to decode and understand instruction formats.

Evaluation Method: Accuracy and speed in completing the puzzle.

Unit 3: Activity: Memory Hierarchy Poster: Creating informative posters or infographics onmemory hierarchy.

Evaluation Method: Clarity of information, presentation and creativity of visual design.

Unit 4: Activity: I/O Troubleshooting Challenge

Evaluation Method: problem identification, feasibility of proposed solutions, and clarity of explanations.

Unit 5: Activity: Case Study on Parallel processing architecture.

Evaluation Method: Understanding of parallel processing concepts and architectures.

III Semester

Course 3: Computer Organization

Credits -1

Lab Experiments

1. Implement a C program to convert a Hexadecimal, octal, and binary number to decimal number vice versa.
2. Implement a C program to perform Binary Addition & Subtraction.
3. Implement a C program to perform Multiplication of two binary numbers.
4. Implement arithmetic micro-operations using logic gates.
5. Implement logic and shift micro-operations using logic gates.
6. Implement a C program to perform Multiplication of two binary numbers (signed) using Booth’s Algorithms.
7. Implement a C program to perform division of two binary numbers (Unsigned) using restoring division algorithm.
8. Implement a C program to perform division of two binary numbers (Unsigned) using non-restoring division algorithm.
9. Write assembly language code for A+B*(C-D) using various instruction formats in MASM or any open-source assembler.
10. Write assembly language code for A+B*C using various addressing modes in MASM or any open-source assembler.

III Semester
Course 8: Operating Systems
Credits -3

Course Objectives

To gain knowledge about various functions of an operating system like memory management, process management, device management, etc.

Course Outcomes:

Upon successful completion of the course, a student will be able to:

1. Demonstrate knowledge and comprehension of operating system functions.
2. Analyze different process scheduling algorithms and apply them to manage processes and threads effectively
3. Create strategies to prevent, detect, and recover from deadlocks, and design solutions for inter-process communication and synchronization problems.
4. Compare and contrast different memory allocation strategies and evaluate their effectiveness
5. Evaluate disk scheduling algorithms while implementing OS security measures

UNIT- I

What is Operating System? History and Evolution of OS, Basic OS functions, Resource Abstraction, Types of Operating Systems– Multiprogramming Systems, Batch Systems, Time Sharing Systems; Operating Systems for Personal Computers, Workstations and Hand-held Devices, Process Control & Real time Systems.

UNIT- II

Processor and User Modes, Kernels, System Calls and System Programs, System View of the Process and Resources, Process Abstraction, Process Hierarchy, Threads, Threading Issues, Thread Libraries; Process Scheduling- Non-Preemptive and Preemptive Scheduling Algorithms.

UNIT III

Process Management: Deadlock, Deadlock Characterization, Necessary and Sufficient Conditions for Deadlock, Deadlock Handling Approaches: Deadlock Prevention, Deadlock Avoidance and Deadlock Detection and Recovery.

Concurrent and Dependent Processes, Critical Section, Semaphores, Methods for Inter process Communication; Process Synchronization, Classical Process Synchronization Problems: Producer-Consumer, Reader-Writer.

UNIT IV

Memory Management: Physical and Virtual Address Space; Memory Allocation Strategies–Fixed and -Variable Partitions, Paging, Segmentation, Virtual Memory.

UNIT V

File and I/O Management, OS security: Directory Structure, File Operations, File Allocation Methods, Device Management, Pipes, Buffer, Shared Memory, Disk Scheduling algorithms.

Text Books:

1. Operating System Principles by Abraham Silberschatz, Peter Baer Galvin and GregGagne (7th Edition) Wiley India Edition.

Reference Books

1. Operating Systems: Internals and Design Principles by Stallings (Pearson)
2. Operating Systems by J. Archer Harris (Author), Jyoti Singh (Author) (TMH)

SUGGESTED CO-CURRICULAR ACTIVITIES & EVALUATION METHODS:

Unit 1: Activity: Case Study on a specific Operating System: highlighting its functions and key features.

Evaluation Method: Case study presentation, depth of understanding of operating system functions, and ability to articulate key concepts.

Unit 2: Activity: Comparison Poster on Scheduling Algorithms

Evaluation Method: Assessment of posters based on content accuracy, clarity of information, visual presentation, and ability to convey key insights.

Unit 3: Activity: Assignment on Dead Lock prevention techniques

Evaluation Method: Understanding, Completion and report.

Unit 4: Activity: Debate on various Memory allocation schemes

Evaluation Method: Debate arguments, ability to counter opposing viewpoints, logical reasoning, and presentation skills.

Unit 5: Activity: Comparative study of various disk scheduling algorithms using real world datasets

Evaluation Method: Analysis methodology, accuracy of results, and presentation of findings and conclusions.

III Semester

Course 8: Operating Systems

Credits -1

List of Experiments:

1. Illustrate the LINUX commands
   a) pwd
   b) mkdir
   c) rmdir
   d) grep
   e) chmod
   f) ls
   g) rm
   h) cp
2. Write a program to calculate average waiting time and turn around time of each process using the following CPU Scheduling algorithm for the given process schedules.
   a) FCFS
   b) SJF
   c) Priority
   d) Round Robin
3. Simulate MVT and MFT memory management techniques
4. Write a program for Bankers Algorithm for Dead Lock Avoidance
5. Implement Bankers Algorithm Dead Lock Prevention.
6. Write a program to simulate Producer-Consumer problem.
7. Simulate all Page replacement algorithms.
   e) FIFO
   f) LRU
   g) LFU
   h) Optimal
8. Simulate Paging Techniques of memory management
9. Simulate the following disk scheduling algorithms
   a) FCFS
   b) SSTF
   c) SCAN
   d) CSCAN

IV Semester

Course 9: Database Management Systems

Credits -3

Learning Objectives:

To familiarize with concepts of database design

Learning Outcomes: On successful completion of the course, students will be able to

1. Differentiate between database systems and file based systems
2. Design a database using ER model
3. Use relational model in database design
4. Use SQL commands for creating and manipulating data stored in databases.
5. Write PL/SQL programs to work with databases.

UNIT- I

Overview of Database Management System: Introduction to data, information, database, database management systems, file-based system, Drawbacks of file-Based System, database approach, Classification of Database Management Systems, advantages of database approach, Various Data Models, Components of Database Management System, three schema architecture of data base, costs and risks of database approach.

UNIT - II

Entity-Relationship Model: Introduction, the building blocks of an entity relationship diagram, classification of entity sets, attribute classification, relationship degree, relationship classification, reducing ER diagram to tables, enhanced entity-relationship model (EER model), generalization and specialization, **IS A** relationship and attribute inheritance, multiple inheritance, constraints on specialization and generalization, advantages of ER modeling.

UNIT - III

Relational Model: Introduction, CODD Rules, relational data model, concept of key, relational integrity, relational algebra, relational algebra operations, advantages of relational algebra,

limitations of relational algebra, relational calculus, tuple relational calculus, domain relational Calculus (DRC), Functional dependencies and normal forms upto 3rd normal form.

UNIT - IV

Structured Query Language: Introduction, Commands in SQL, Data Types in SQL, Data Definition Language, Selection Operation, Projection Operation, Aggregate functions, Data Manipulation Language, Table Modification Commands, Join Operation, Set Operations, View, Sub Query.

UNIT - V

PL/SQL: Introduction, Shortcomings of SQL, Structure of PL/SQL, PL/SQL Language Elements,Data Types, Operators Precedence, Control Structure, Steps to Create a PL/SQL, Program, Iterative Control, Procedure, Function, Database Triggers, Types of Triggers.

Text Books:

1. Operating System Principles by Abraham Silberschatz, Peter Baer Galvin and GregGagne (7th Edition) Wiley India Edition.

Reference Books

1. Database Management Systems by Raghu Ramakrishnan, McGrawhill
2. Principles of Database Systems by J. D. Ullman
3. Fundamentals of Database Systems by R. Elmasri and S. Navathe
4. SQL: The Ultimate Beginners Guide by Steve Tale.

SUGGESTED CO-CURRICULAR ACTIVITIES & EVALUATION METHODS:

Unit 1: Activity: Seminar Presentation on Database Management Systems

Evaluation Method: Depth of research, clarity of explanations, ability to address questions and engage the audience.

Unit 2: Activity: Case Study on EER model

Evaluation Method: Identification of inheritance relationships, effective use of generalization and specialization, and adherence to constraints.

Unit 3: Activity: Exercise on Normalization: Assign students a set of unnormalized tables and have them normalize the tables to third normal form

Evaluation Method: Normalized table designs, identification of functional dependencies,adherence to normalization rules, and elimination of anomalies.

Unit 4: Activity: Competition on SQL Query Writing

Evaluation Method: Query correctness, efficiency, proper use of SQL commands, ability to handle complex scenarios, and creativity in query formulation.

Unit 5: Activity: Peer Review of PL/SQL code

Evaluation Method: Peer evaluation of code quality, adherence to coding standards, proper use of language elements, and logic.

IV Semester

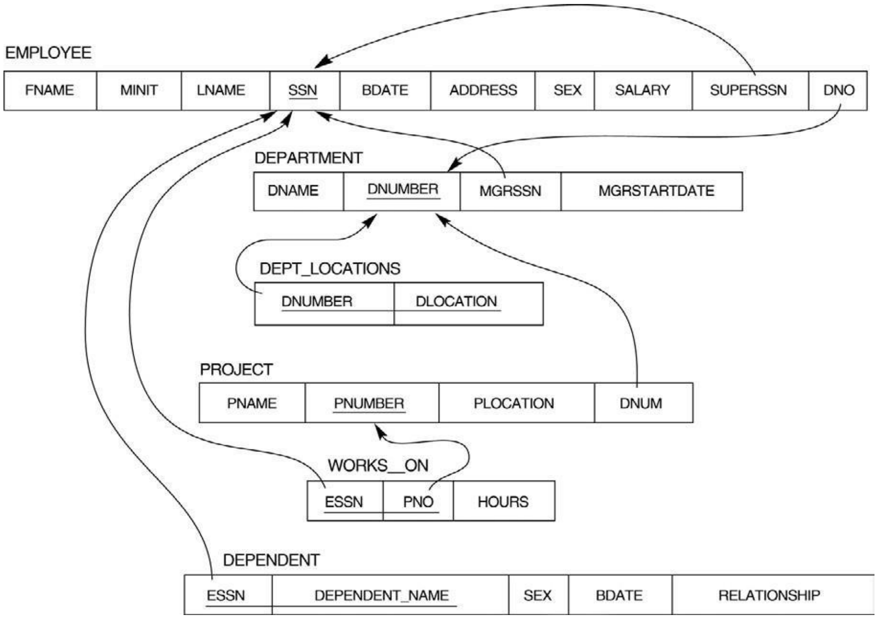
Course 9: Database Management Systems

Credits -1

List of Experiments:

1. Draw ER diagram for hospital administration
2. Creation of college database and establish relationships between tables
3. Relational database schema of a company is given in the following figure.

Relational Database Schema - COMPANY



Questions to be performed on above schema

1. Create above tables with relevant Primary Key, Foreign Key and other constraints
2. Populate the tables with data
3. Display all the details of all employees working in the company.
4. Display ssn, lname, fname, address of employees who work in department no 7.
5. Retrieve the Birthdate and Address of the employee whose name is 'Franklin T. Wong'
6. Retrieve the name and salary of every employee
7. Retrieve all distinct salary values
8. Retrieve all employee names whose address is in ‘Bellaire’
9. Retrieve all employees who were born during the 1950s
10. Retrieve all employees in department 5 whose salary is between 50,000 and 60,000(inclusive)

11. Retrieve the names of all employees who do not have supervisors
12. Retrieve SSN and department name for all employees
13. Retrieve the name and address of all employees who work for the 'Research' department
14. For every project located in 'Stafford', list the project number, the controlling department number, and the department manager's last name, address, and birth date.
15. For each employee, retrieve the employee's name, and the name of his or her immediate supervisor.
16. Retrieve all combinations of Employee Name and Department Name
17. Make a list of all project numbers for projects that involve an employee whose last name is 'Narayan' either as a worker or as a manager of the department that controls the project.
18. Increase the salary of all employees working on the 'Product X' project by 15%. Retrieve employee name and increased salary of these employees.
19. Retrieve a list of employees and the project name each works in, ordered by the employee's department, and within each department ordered alphabetically by employeefirst name.
20. Select the names of employees whose salary does not match with salary of any employee in department 10.
21. Retrieve the employee numbers of all employees who work on project located in Bellaire, Houston, or Stafford.
22. Find the sum of the salaries of all employees, the maximum salary, the minimum salary,and the average salary. Display with proper headings.
23. Find the sum of the salaries and number of employees of all employees of the ‘Marketing’ department, as well as the maximum salary, the minimum salary, and the average salary in this department.
24. Select the names of employees whose salary is greater than the average salary of all employees in department 10.
25. Delete all dependents of employee whose ssn is ‘123456789’.
26. Perform a query using alter command to drop/add field and a constraint in Employee table.

IV Semester
Course 10: Object Oriented Software Engineering
Credits -3

Course Objective:

To introduce Object-oriented software engineering (OOSE) - which is a popular technical approach to analyzing, designing an application, system, or business by applying the object- oriented paradigm and visual modeling.

Course Outcomes:

Upon successful completion of the course, a student will be able to:

1. Understand and apply the fundamental principles of Object-Oriented Programming (OOP) concepts and Unified Modeling Language (UML) basics, in the development of software solutions.
2. Analyze and specify software requirements, develop use cases and scenarios, apply object-oriented analysis and design (OOAD) principles
3. Familiar with the concept of test-driven development (TDD) and its practicalimplementation
4. Analyze and Evaluate Software Maintenance and Evolution Strategies
5. Apply Advanced Object-Oriented Software Engineering Concepts

UNIT-I

Introduction to Object-Oriented Programming: Overview of software engineering, Introduction to Object-Oriented Programming (OOP) concepts (classes, objects, inheritance, polymorphism), Unified Modelling Language (UML) basics, Introduction to software development process and software development life cycle (SDLC).

UNIT-II

Requirements Analysis and Design: Requirements analysis and specification, Use cases and scenarios, Object-oriented analysis and design (OOAD), Design patterns, UML modelling techniques (class diagrams, sequence diagrams, state machine diagrams, activity diagrams)

UNIT-III

Software Construction and Testing: Software construction basics, Object-oriented design principles, Object-oriented programming languages (Java, C++, Python), Software testing basics (unit testing, integration testing, system testing), Test-driven development (TDD)

UNIT-IV

Software Maintenance and Evolution: Software maintenance basics, refactoring techniques Software version control, Code review and inspection, Software evolution and reengineering

UNIT-V

Advanced Topics in Object-Oriented Software Engineering: Model-driven engineering (MDE), Aspect-oriented programming (AOP), Component-based software engineering (CBSE), Service-oriented architecture (SOA), Agile software development and Scrum methodologies.

Text Book(s)

1. An Introduction to Object-Oriented Analysis and Design and the Unified Process, 3rd Edition, Craig Larman, Prentice-Hall.
2. Programming in Java by Sachin Malhotra, Oxford University Press

Reference Books

1. Requirements engineering: processes and techniques, G.Kotonya and, I.Sommerville, 1998, Wiley
2. Design Patterns, E.Gamma, R. Helm, R. Johnson, and J. Vlissides
3. The Unified Modeling Language Reference Manual, J. Rumbaugh, I.Jacobson and G. Booch, Addison Wesley

SUGGESTED CO-CURRICULAR ACTIVITIES & EVALUATION METHODS:

Unit 1: Activity: Group Activity: Design and implement a small OOP project

Evaluation Method: Presentation evaluation rubric, Project evaluation based on OOP principles.

Unit 2: Activity: Use Case Scenario Presentation & Peer Activity: Review and provide feedback on each other's use case diagrams

Evaluation Method: Presentation evaluation rubric, Peer feedback assessment.

Unit 3: Activity: Poster Presentation: Illustrate TDD principles and benefits

Evaluation Method: Poster presentation evaluation

Unit 4: Activity: Peer Activity: Analyze and discuss different maintenance strategies

Evaluation Method: Peer discussion participation evaluation

Unit 5: Activity: Seminar on Design Patterns

Evaluation Method: Depth of research, clarity of explanations, ability to address questions and engage the audience.

IV Semester
Course 10: Object Oriented Software Engineering

Credits -1

Suggested Software Tools: StarUML/UMLGraph/Topcased/Umberollo/ArgoUML/ Eclipse IDE, Visual Paradigm for UML/Rational Software Architect/Any other Open Source Tool

List of Experiments:

Select domain of interest (e.g. College Management System) and identify multi-tier software application to work on (e.g. Online Fee Collection). Analyze, design and develop this application using OOSE approach:

1. Develop an IEEE standard SRS document. Also develop risk management and project plan (Gantt chart).
2. Understanding of System modeling: Data model i.e. ER – Diagram and draw the ER Diagram with generalization, specialization and aggregation of specified problem statement
3. Understanding of System modeling: Functional modeling: DFD level 0 i.e. Context Diagram and draw it
4. Understanding of System modeling: Functional modeling: DFD level 1 and DFD level 2 and draw it.
5. Identify use cases and develop the use case model.
6. Identify the business activities and develop an UML Activity diagram.
7. Identity the conceptual classes and develop a domain model with UML Class diagram.
8. Using the identified scenarios find the interaction between objects and represent them using UML Interaction diagrams.
9. Draw the state chart diagram.
10. Identify the user interface, domain objects, and technical services. Draw the partial layered, logical architecture diagram with UML package diagram notation.
11. Implement the technical services layer.
12. Implement the domain objects layer.
13. Implement the user interface layer.
14. Draw component and deployment diagrams.

IV Semester

Course 11: Data Communication and Computer Networks

Credits -3

Course Objectives

To provide students with a comprehensive understanding of networking principles, protocols, and technologies, enabling them to design, analyze, and evaluate efficient and reliable network solutions.

Course Outcomes

Upon successful completion of the course, a student will be able to:

1. Understand and apply network applications, hardware, software, and reference models for network communication.
2. Design and analyze data link layer protocols, multiple access protocols, and wireless LAN technologies.
3. Design routing algorithms, congestion control algorithms, and evaluate network layer protocols for internetworking.
4. Analyze transport service, transport protocols, and evaluate UDP and TCP in the internet.
5. Understand and evaluate application layer protocols, including DNS, email, WWW, and network management protocols.

UNIT-I

INTRODUCTION: Network applications, network hardware, network software, reference models: OSI, TCP/IP, Internet, Connection oriented network - X.25, frame relay.

THE PHYSICAL LAYER: Theoretical basis for communication, guided transmission media, wireless transmission, the public switched telephone networks, mobile telephone system.

UNIT-II

THE DATA LINK LAYER: Design issues, error detection and correction, elementary data link protocols, sliding window protocols, example data link protocols - HDLC, the data link layer on the internet.

THE MEDIUM ACCESS SUBLAYER: Channel allocations problem, multiple access protocols, Ethernet, Data Link Layer switching, Wireless LAN, Broadband Wireless, Bluetooth.

UNIT-III

THE NETWORK LAYER: Network layer design issues, routing algorithms, Congestion control algorithms, Internetworking, the network layer in the internet (IPv4 and IPv6), Quality of Service.

UNIT-IV

THE TRANSPORT LAYER: Transport service, elements of transport protocol, SimpleTransport Protocol, Internet transport layer protocols: UDP and TCP.

UNIT-V

THE APPLICATION LAYER: Domain name system, electronic mail, World Wide Web: architectural overview, dynamic web document and http.

APPLICATION LAYER PROTOCOLS: Simple Network Management Protocol, File Transfer Protocol, Simple Mail Transfer Protocol, Telnet.

Text Book(s)

1. S. Tanenbaum (2003), Computer Networks, 4th edition, Pearson Education/ PHI, New Delhi, India

Reference Books

2. Behrouz A. Forouzan (2006), Data communication and Networking, 4th Edition, Mc Graw-Hill, India.
3. Kurose, Ross (2010), Computer Networking: A top down approach, Pearson Education, India.

SUGGESTED CO-CURRICULAR ACTIVITIES & EVALUATION METHODS:

Unit 1: Activity: Hands-on exercises to configure network applications

Evaluation Method: Practical skills in configuring network applications, hardware, and software.

Unit 2: Activity: Protocol Design and Simulation using simulation tools like NS-3 or Cisco Packet Tracer.

Evaluation Method: Students' ability to design and simulate data link layer protocols and multiple access protocols

Unit 3: Activity: Guest Lectures and Workshops on routing algorithms, congestion control, and network layer protocols.

Evaluation Method: Students' participation and understanding demonstrated in guest lectures and workshop

Unit 4: Activity: Network Monitoring and Traffic Analysis using tools like Wireshark

Evaluation Method: Understanding of transport protocols through their analysis ofnetwork traffic and identification of UDP and TCP behavior

Unit 5: Activity: Group Projects on Network Application Development

Evaluation Method: Group Project Presentations

IV Semester

Course 11: Data Communication and Computer Networks

Credits -1

List of Experiments:

1. Understanding various network tools in Windows and Linux
2. Study different types of Network devices and Cables
3. Building a Local Area Network
4. Concept of Network IP Address
5. Introduction to Network Simulator – Packet Tracer (PT)
6. Configuration of a Router using Packet Tracer
7. Implementation of a Network using Packet Tracer
8. Implementation of Static Routing using Packet Tracer
9. Implementation of RIP using Packet Tracer
10. Implementation of OSPF using Packet Tracer
11. Implement DNS using packet tracer
12. Implementation of a VLAN using Packet Tracer

V Semester

Course 12: Web Interface Designing Technologies

Credits -3

Learning Objectives:

To enable students to understand web architecture, develop aesthetic websites, create static and dynamic web pages, implement user interactivity, and gain proficiency in installing and utilizing WordPress and plugins

Learning Outcomes: On successful completion of the course, students will be able to

1. Understand and appreciate the web architecture and services along with its basic building blocks
2. Gain knowledge about various components of a website related to aesthetics
3. Demonstrate skills regarding creation of a static website and addition of dynamic behaviorto a website
4. Get experience on making user-interactive web pages.
5. Learn how to install word press and gain the knowledge of installing various plugins to usein their websites.

UNIT - I

HTML: Introduction to web designing, difference between web applications and desktop applications, introduction to HTML, HTML structure, elements, attributes, headings, paragraphs, images, tables, lists, blocks, symbols, embedding multi-media components in HTML, HTML forms

UNIT – II

CSS: CSS home, introduction, syntax, CSS combinators, colors, background, borders, margins, padding, height/width, text, fonts, tables, lists, position, overflow, float, pseudo class, pseudo elements, opacity, tool tips, image gallery, CSS forms, CSS counters.

UNIT – III

Java Script: What is DHTML, JavaScript, basics, variables, operators, statements, string manipulations, mathematical functions, arrays, functions. objects, regular expressions, exception handling.

UNIT-IV

Client-Side Scripting: Accessing HTML form elements using Java Script object model, basic data validations, data format validations, generating responsive messages, opening windows using java script, different kinds of dialog boxes, accessing status bar using java script, embedding basic animative features using different keyboard and mouse events.

UNIT – V

Word press: Introduction to word press, features, and advantages, installing and configuring word press and understanding its admin panel (demonstration only), working with posts, managing pages, working with media - Adding, editing, deleting media elements, working with widgets, using menus, working with themes, defining users, roles and profiles, adding external links, extending word press with plug-ins.

Text Book(s)

1. Chris Bates, Web Programming Building Internet Applications, Second Edition, Wiley (2007)
2. Paul S.WangSanda S. Katila, an Introduction to Web Design plus Programming, Thomson (2007).

Reference Books

1. Head First HTML and CSS, Elisabeth Robson, Eric Freeman, O'Reilly Media Inc.
2. An Introduction to HTML and JavaScript: for Scientists and Engineers, David R. Brooks. Springer, 2007
3. Schaum's Easy Outline HTML, David Mercer, Mcgraw Hill Professional.
4. Word press for Beginners, Dr.Andy Williams.
5. Professional word press, Brad Williams, David damstra, Hanstern.

SUGGESTED CO-CURRICULAR ACTIVITIES & EVALUATION METHODS:

Unit 1: Activity: Infographic explaining the necessity to have a web site for each of the agenciessuch as hotels, hospitals, supermarkets, and educational institutions.

Evaluation Method: Assess the accuracy, visual design, clarity, creativity, use of visual elements, presentation of the infographic explaining the necessity of a website for different agencies.

Unit 2: Activity: Seminar though PPT on various Look and Feel components that websites relatedto different agencies

Evaluation Method: Content knowledge, organization, clarity, presentation skills, visualaids, audience engagement

Unit 3: Activity: Code snippets Challenge.

Evaluation Method: Accuracy, functionality, efficiency, code readability, and problem-solving approach of the JavaScript code snippets

Unit 4: Activity: Group discussion on different kinds of web forms that take and validate userinput using java script validations

Evaluation Method: Active participation, knowledge sharing, critical thinking, and demonstration of different web forms and JavaScript validations

Unit 5: Activity: Creation of Personal website using wordpress

Evaluation Method: Design aesthetics, functionality, user interactivity, contentorganization, and utilization of plugins.

V Semester
Course 12: Web Interface Designing Technologies
Credits -1

List of Experiments:

1. Create an HTML document with the following formatting options:

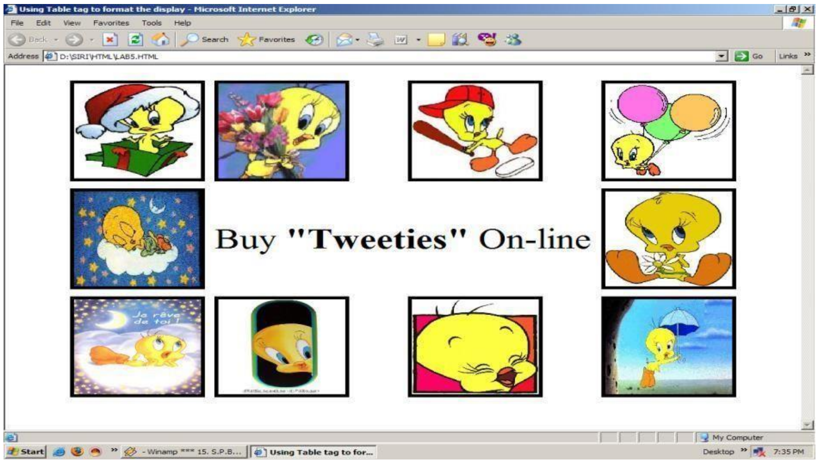
   (a) Bold, (b) Italics, (c) Underline, (d) Headings (Using H1 to H6 heading styles), (e) Font (Type, Size and Color), (f) Background (Colored background/Image in background), (g) Paragraph, (h) Line Break, (i) Horizontal Rule, (j) Pre tag

2. Create an HTML document which consists of:

   (a) Ordered List (b) Unordered List (c) Nested List (d) Image

3. Create a Table with four rows and five columns. Place an image in one column.

4. Using “table” tag, align the images as follows:


Using Table tag to format the display - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address D:\SIRI\HTML\LAB5.HTML
Go Links
Buy "Tweeties" On-line
My Computer
Start - Winamp *** 15. S.P.B... Using Table tag to for...
Desktop 7:35 PM

5. Create a menu form using html.

6. Style the menu buttons using CSS.

7. Create a form using HTML which has the following types of controls:
   (a) Text Box (b) Option/radio buttons (c) Check boxes (d) Reset and Submit buttons

8. Embed a calendar object in your web page.

9. Create a form that accepts the information from the subscriber of a mailing system.

Word press:

10. Installation and configuration of word press

11. Access admin panel and manage posts

12. Access admin panel and manage pages

13. Add widgets and menus

14. Create users and assign roles

15. Create a site and add a theme to it

V Semester
Course 13: Web Applications Development using PHP & MYSQL
Credits -3

Learning Objectives:

To enable students to understand open-source tools to create dynamic web pages, implement user interactivity, and gain proficiency in developing web sites

Learning Outcomes: On successful completion of the course, students will be able to

1. Write simple programs in PHP.
2. Understand how to use regular expressions, handle exceptions, and validate data using PHP.
3. Apply In-Built functions and Create User defined functions in PHP programming.
4. Write PHP scripts to handle HTML forms.
5. Know how to use PHP with a MySQL database and can write database driven web pages.

UNIT-I

The building blocks of PHP: Variables, Data Types, Operators and Expressions, Constants. **Flow Control Functions in PHP**: Switching Flow, Loops, Code Blocks and Browser Output.**Working with Functions:** Creating functions, Calling functions, Returning the values from User- Defined Functions, Variable Scope, Saving state between Function calls with the static statement,arguments of functions

UNIT-II

Working with Arrays: Creating Arrays, Some Array-Related Functions.

Working with Objects: Creating Objects, Accessing Object Instances, **Working with Strings**, **Dates and Time:** Formatting strings with PHP, Manipulating Strings with PHP, Using Date and Time Functions in PHP.

UNIT-III

Working with Forms: Creating Forms, Accessing Form Input with User defined Arrays, Combining HTML and PHP code on a single Page, Using Hidden Fields to save state, Redirectingthe user, Sending Mail on Form Submission, and **Working with File Uploads**, Managing files onserver, **Exception handling.**

UNIT-IV

Working with Cookies and User Sessions: Introducing Cookies, setting a Cookie with PHP, Session Function Overview, starting a Session, working with session variables, passing session IDs in the Query String, Destroying Sessions and Unsetting Variables, Using Sessions in an Environment with Registered Users.

UNIT-V

Interacting with MySQL using PHP: MySQL Versus MySQLi Functions, connecting to MySQL with PHP, Working with MySQL Data. Planning and Creating Database Tables, Creating Menu, Creating Record Addition Mechanism, Viewing Records, Creating the Record Deletion Mechanism.

Text Book(s)

1. Julie C. Meloni, SAMS Teach yourself PHP MySQL and Apache, Pearson Education (2007).
2. Steven Holzner , PHP: The Complete Reference, McGraw-Hill

Reference Books

1. Robin Nixon, Learning PHP, MySQL, JavaScript, CSS & HTML5, Third Edition O'reilly, 2014
2. Xue Bai Michael Ekedahl, The web warrior guide to Web Programming, Thomson (2006).

SUGGESTED CO-CURRICULAR ACTIVITIES & EVALUATION METHODS:

Unit 1: Activity: Infographic explanation of client-server architecture and different server-side scripting languages.

Evaluation Method: Assess the accuracy, visual design, clarity, creativity, use of visual elements, presentation of the infographic explaining the benefits of server-side scripting languages.

Unit 2: Activity: Presentation on various open-source frameworks available in LAMP model

Evaluation Method: Content knowledge, organization, clarity, presentation skills, visual aids, audience engagement

Unit 3: Activity: Code snippets Challenge.

Evaluation Method: Accuracy, functionality, efficiency, code readability, and problem-solving approach of the PHP code snippets

Unit 4: Activity: Group discussion on Session Management in PHP

Evaluation Method: Active participation, knowledge sharing, critical thinking, and demonstration of Session Management

Unit 5: Activity: Hands-on Lab Session on MYSQL Queries

Evaluation Method: Lab Performance and Correctness of solution Implementation

V Semester
Course 13: Web Applications Development using PHP & MYSQL
Credits -1

List of Experiments:

1. Write a PHP program to Display “Hello”
2. Write a PHP Program to display the today’s date.
3. Write a PHP program to display Fibonacci series.
4. Write a PHP Program to read the employee details.
5. Write a PHP program to prepare the student marks list.
6. Create student registration form using text box, check box, radio button, select, submit button. And display user inserted value in new PHP page.
7. Create Website Registration Form using text box, check box, radio button, select, submit button. And display user inserted value in new PHP page.
8. Write PHP script to demonstrate passing variables with cookies.
9. Write a PHP script to connect MySQL server from your website.
10. Write a program to keep track of how many times a visitor has loaded the page.
11. Write a PHP application to perform CRUD (Create, Read, Update and Delete) operations on a database table.
12. Create a web site using any open-source framework built on PHP and MySQL – It is a team activity wherein students are divided into multiple groups and each group comes up with their own website with basic features.

V Semester
Course 14 A: Internet of Things
Credits -3

Learning Objectives:

To enable students to understand basic IoT constructs, create IoT solutions to real world problems using IoT

Learning Outcomes: On successful completion of the course, students will be able to

1. Understand various concepts, terminologies and applications of IoT
2. Learn how to build IoT devices with development boards
3. Understand various Wireless protocols for IoT
4. Learn how to use various sensors and actuators & develop IoT solutions using Arduino
5. Develop and Connect IoT with Cloud Platforms.

UNIT - I

Fundamentals of IoT: Introduction, Definitions & Characteristics of IoT, IoT Architectures, Physical & Logical Design of IoT, Enabling Technologies in IoT, History of IoT, About Things in IoT, The Identifiers in IoT, About the Internet in IoT, IoT frameworks, IoT and M2M.

Applications of IoT: Home Automation, Smart Cities, Energy, Retail Management, Logistics, Agriculture, Health and Lifestyle, Industrial IoT, Legal challenges, IoT design Ethics, IoT in Environmental Protection.

UNIT - II

Sensors Networks : Definition, Types of Sensors, Types of Actuators, Examples and Working, IoT Development Boards: Arduino IDE and Board Types, RaspberriPi Development Kit, RFID Principles and components, Wireless Sensor Networks: History and Context, The node, Connecting nodes, Networking Nodes, WSN and IoT.

Unit - III

Wireless Technologies for IoT: WPAN Technologies for IoT: IEEE 802.15.4, Zigbee, HART, NFC, Z-Wave, BLE, Bacnet and Modbus.

IP Based Protocols for IoT: IPv6, 6LowPAN, LoRA, RPL, REST, AMPQ, CoAP, MQTT. Edge connectivity and protocols.

Unit - IV

Arduino Simulation Environment: Arduino Uno Architecture, Setting up the IDE, Writing Arduino Software, Arduino Libraries, Basics of Embedded C programming for Arduino, Interfacing LED, push button and buzzer with Arduino, Interfacing Arduino with LCD.

Sensor & Actuators with Arduino: Overview of Sensors working, Analog and Digital Sensors, Interfacing of Temperature, Humidity, Motion, Light and Gas Sensors with Arduino, Interfacing of Actuators with Arduino, Interfacing of Relay Switch and Servo Motor with Arduino.

Unit - V

Developing IOT's: Implementation of IoT with Arduino, Connecting and using various IoT Cloud Based Platforms such as Blynk, Thingspeak, AWS IoT, Google Cloud IoT Core etc. Cloud Computing, Fog Computing, Privacy and Security Issues in IoT.

Text Book(s)

1. Internet of Things - A Hands-on Approach, ArshdeepBahga and Vijay Madisetti, Universities Press, 2015, ISBN: 9788173719547
2. Sudip Mishra, Anandarup Mukherjee, Arijit Roy: Introduction to IOT, Cambridge University Press.
3. Internet of Things- Dr Surya Durbha & Dr Jyoti Joglekar, Oxford University Press

Reference Books

1. Daniel Minoli, — "Building the Internet of Things with IPv6 and MIPv6: The Evolving World of M2M Communications", ISBN: 978-1-118-47347-4, Willy Publications
2. Pethuru Raj and Anupama C. Raman, "The Internet of Things: Enabling Technologies, Platforms, and Use Cases", CRC Press

SUGGESTED CO-CURRICULAR ACTIVITIES & EVALUATION METHODS:

Unit 1: Activity: Seminar on various applications of IoT through PPT

Evaluation Method: Content knowledge, organization, clarity, presentation skills, visualaids, audience engagement

Unit 2: Activity: Hands-on Lab activity on Arduino Development

Evaluation Method: Lab Performance and Correctness of Circuit Implementation

Unit 3: Activity: Group discussion on Future Wireless Technologies.

Evaluation Method: Active participation, knowledge sharing, critical thinking, and demonstration of different wireless technologies for IoT

Unit 4: Activity: Peer activity on different types of Sensors

Evaluation Method: Peer evaluation of working principle of Sensor, use-cases of sensors.

Unit 5: Activity: Guest Lecture or Expert talk on Cloud based IoT platforms

Evaluation Method: Active Participation, Post Talk report presentation

V Semester

Course 14 A: Internet of Things

Credits -1

List of Experiments:

1. Understanding Arduino UNO Board and Components
2. Installing and work with Arduino IDE
3. Blinking LED sketch with Arduino
4. Simulation of 4-Way Traffic Light with Arduino
5. Using Pulse Width Modulation
6. LED Fade Sketch and Button Sketch
7. Analog Input Sketch (Bar Graph with LEDs and Potentiometre)
8. Digital Read Serial Sketch (Working with DHT/IR/Gas or Any other Sensor)
9. Working with Adafruit Libraries in Arduino
10. Spinning a DC Motor and Motor Speed Control Sketch
11. Working with Shields
12. Design APP using Blink App or Things peak API and connect it LED bulb.
13. Design APP Using Blynk App and Connect to Temperature, magnetic Sensors.

V Semester
Course 14 B: Foundations of Data Science

Credits -3

Learning Objectives:

To enable students to develop IoT solutions for real-world problems

Learning Outcomes: On successful completion of the course, students will be able to

1. Identify the need for data science and understand various data collection strategies
2. Understand about NoSQL and Descriptive Statistics
3. Apply Numpy methods to process the data in an array.
4. Summarize and Compute Descriptive Statistics using Pandas.
5. Apply powerful data manipulations visualization using Pandas

UNIT-I

Introduction to Data Science: Need for Data Science – What is Data Science - Evolution of Data Science, Data Science Process – Business Intelligence and Data Science – Prerequisites for a Data Scientist – Tools and Skills required. Applications of Data Science in various fields – Data Security Issues.

Data Collection Strategies, Data Pre-Processing Overview, Data Cleaning, Data Integration and Transformation, Data Reduction, Data Discretization, Data Munging, Filtering

UNIT-II

Descriptive Statistics – Mean, Standard Deviation, Skewness and Kurtosis; Box Plots – Pivot Table – Heat Map – Correlation Statistics –ANOVA.

No-SQL: Document Databases, Wide-column Databases and Graphical Databases.

UNIT-III

Python for Data Science –Python Libraries, Python integrated Development Environments (IDE) for Data Science, **NumPy Basics:** Arrays and Vectorized Computation- The NumPy ndarray-

Creating ndarrays- Data Types for ndarrays- Arithmetic with NumPy Arrays- Basic Indexing and Slicing - Boolean Indexing-Transposing Arrays and Swapping Axes.

Universal Functions: Fast Element-Wise Array Functions- Mathematical and Statistical Methods- Sorting- Unique and Other Set Logic.

UNIT-IV

Introduction to pandas Data Structures: Series, Data Frame and Essential Functionality: Dropping Entries- Indexing, Selection, and Filtering- Function Application and Mapping- Sortingand Ranking.

Summarizing and Computing Descriptive Statistics- Unique Values, Value Counts, and Membership. Reading and Writing Data in Text Format.

UNIT-V

Data Cleaning and Preparation: Handling Missing Data - Data Transformation: Removing Duplicates, Transforming Data Using a Function or Mapping, Replacing Values, Detecting and Filtering Outliers-

Plotting with pandas: Line Plots, Bar Plots, Histograms and Density Plots, Scatter or Point Plots.

Text Book(s)

1. Y. Daniel Liang, “Introduction to Programming using Python”, Pearson, 2012.
2. Wes McKinney, “Python for Data Analysis: Data Wrangling with Pandas, NumPy,and IPython”, O’Reilly, 2nd Edition, 2018.

Reference Books

1. Sanjeev Wagh, Manisha Bhende, Anuradha Thakare, ‘Fundamentals of Data Science, CRC Press, 1st Edition, 2022
2. Jake VanderPlas, “Python Data Science Handbook: Essential Tools for Working with Data”, O’Reilly, 2017.

SUGGESTED CO-CURRICULAR ACTIVITIES & EVALUATION METHODS:

Unit 1: Activity: Seminar on Role of Data Science in Politics

Evaluation Method: Content knowledge, organization, clarity, presentation skills, visualaids, audience engagement

Unit 2: Activity: Exercises on Descriptive Statistics

Evaluation Method: Problem Solving, Accuracy

Unit 3: Activity: Hands-on Lab using Numpy

Evaluation Method: Lab Performance and Correctness of solution Implementation

Unit 4: Activity: Hands-on Lab Activity on Pandas

Evaluation Method: Lab Performance and Correctness of solution Implementation.

Unit 5: Activity: Group Activity to visualize college performance records using various plots

Evaluation Method: Active Participation, Post Talk report presentation

V Semester

Course 14 B : Foundations of Data Science

Credits -1

List of Experiments:

1. Study on various python IDEs for Data Science
2. Create NumPy arrays from Python Data Structures, Intrinsic NumPy objects and Random Functions.
3. Manipulation of NumPy arrays- Indexing, Slicing, Reshaping, Joining and Splitting.
4. Computation on NumPy arrays using Universal Functions and Mathematical methods.
5. Create Pandas Series and Data Frame from various inputs.
6. Import any CSV file to Pandas Data Frame and perform the following:
   a. Visualize the first and last 10 records
   b. Get the shape, index and column details
   c. Select/Delete the records (rows)/columns based on conditions.
   d. Perform ranking and sorting operations.
   e. Do required statistical operations on the given column
7. Import any CSV file to Pandas Data Frame and perform the following:

a. Handle missing data by detecting and dropping/ filling missing values.

b. Transform data using apply () and map() method.

c. Detect and filter outliers.

d. Perform Vectorized String operations on Pandas Series.

e. Visualize data using Line Plots, Bar Plots, Histograms, Density Plots and Scatter Plots.

V Semester

Course 15 A: IoT Applications Development and Programming

Theory 03 hours /Week Credits -3

Learning Objectives:

To enable students to develop IoT solutions for real-world problems

Learning Outcomes: On successful completion of the course, students will be able to

1. Understand the Basic Concepts of Internet of Things
2. Learn various Sensors and their associative protocols
3. Learn the Single Board Computers for development of IoT
4. Build the IoT devices with the Node-RED without Complex coding
5. Develop various IoT real-time applications

UNIT-I

Overview of the Internet of Things (IoT) and Sensors: Sensors - Energy-based, Signal Output, Mode of Operation, Electronic Sensors. Connectivity - Bluetooth, Zigbee, Wi-Fi, LoRa, Wired Communication. Machine Intelligence, Active Management, Sensor Fusion, Smart Devices-Human-Computer Interaction, Context Awareness, Actuators, IoT and Smart City Applications- Automobile Sensors, Smart Home Sensors, Smart Transportation Sensors.

UNIT-II

IoT Sensors and Their Interfacing Protocols: Vision and Imaging Sensors- Line Scan Cameras, 3D Depth Cameras, **Sensors That Measure Temperature**-Thermocouples, Resistance Temperature Detector (RTD), Temperature Thermistor Sensors, Semiconductor Temperature Sensors, Radiation Sensors; Proximity Sensors, Pressure Sensors, Position Sensors, Photoelectric Sensors, Particle Sensors, Types of Particle Sensors-Metal Detectors, Level Sensors, Leak Detectors, Humidity Sensors, Gas and Chemical Sensors, Gas Detectors,Carbon Monoxide (MQ7) Detectors, Flame Detectors, **Sensor Communication Protocols**

UNIT-III

Programming Single Board Computers: Arduino Programming, Raspberry Pi-Basic functionality of Raspberry Pi B+ board, setting up the board, configuration and use, Basics of Linux and its use, Introduction to Raspberry Pi GPIO Access, Interfacing DHT, Interfacing Picam to Raspberry Pi zero w, Pi Camera Specifications, Pi Camera Access, Interfacing PIR Sensor **Python:**

File Concepts, Spreadsheet Concepts, Communication Concepts, Wired and WirelessProgramming Concepts

UNIT-IV

Node-RED: Node-RED Features, Installation of Node-RED, Node-RED Architecture, Node- RED Flow Editor, Basic Function Nodes, Node-RED Library, Node-RED Applications; MQTT Protocols, Google Sheets Programming (gspread), Firebase Programming, Matplotlib- Getting Started, Bar Graphs, Scatter Plot, Spectrum Representation, Coherence of Two Signals, Cross- Correlation Graph, Autocorreleation Graph, Changing Figure Size in Different Units, Scale Pie Charts, Style Sheets- FiveThirtyEight Style Sheet, Solarized Light Style Sheet.

UNIT-V

Wireless Connectivity in IoT: Introduction, Low-Power Wide-Area Networks (LPWANs),RFID Protocol, XBEE Radios with Arduino, Bluetooth with Arduino, Arduino with a GSMModem, Arduino with Firebase Cloud Connectivity

The Internet of Things through the Raspberry Pi: Introduction, Cluster Computing with Raspberry Pi Zero W-Message Passing Interface (MPI), Networking with RP is for Simple MPI Scripts, Simple MPI Programming

Text Book(s)

1. **Internet of Things Using Single Board Computers**, *G. R. Kanagachidambaresan,* Apress, 2022.
2. **Practical Node-RED Programming**, *Taiji Hagino*, Packt Publishing, 2021

Reference Books

1. **Internet of Things Programming Projects: Build modern IoT solutions with the Raspberry Pi 3 and Python,** *Colin Dow***,** Packt Publishing, 2021
2. **Programming the Internet of Things: An Introduction to Building Integrated, Device-to-Cloud IoT Solutions,** *Andy King***,** O’Reilly Media, 2021

SUGGESTED CO-CURRICULAR ACTIVITIES & EVALUATION METHODS:

Unit 1: Activity: Case Study Presentation on Smart City IoT realization

Evaluation Method: Content knowledge, organization, clarity, presentation skills, visualaids, audience engagement

Unit 2: Activity: Poster Presentation for various kinds of Sensors

Evaluation Method: Creative & informative posters or infographics on Sensors

Unit 3: Activity: Hands-on Lab using RPi.

Evaluation Method: Lab Performance and Correctness of solution Implementation

Unit 4: Activity: Hands-on Lab Activity on Node-RED

Evaluation Method: Lab Performance and Correctness of solution Implementation.

Unit 5: Activity: Guest Lecture or Expert talk on Cloud based IoT platforms

Evaluation Method: Active Participation, Post Talk report presentation

V Semester
Course 15 A: IoT Applications Development and Programming

Credits -1

List of Experiments:

1. Write a program to switch light on when the input is 1 and switch the light off when the input is 0 using Raspberry pi
2. Install Node-RED and Flow-based Programming Development Environment
3. Create Basic Flows with Major Nodes
4. Develop a Node-Red Flow for various Case Studies
5. Implement Node-RED in the Cloud Calling a Web API from Node-RED
6. Create a To Do Application with Node-RED Handling Sensor Data on the Raspberry Pi
7. Develop a Dashboard with various 2D Graphs with Matplotlib
8. Install MySQL database in Raspberry pi.
9. Write a program to work with basic MySQL queries by fetching data from database in Raspberry pi.
10. Arduino with Firebase Cloud Connectivity
11. Visualize Data by Creating a Server-side Application in the Firebase

V Semester

Course 15 B : Application Development using Python

Credits -3

Learning Objectives:

To enable students to develop IoT solutions for real-world problems

Learning Outcomes: On successful completion of the course, students will be able to

1. Examine Python syntax and semantics and be fluent in the use of Python flow control and functions.
2. Demonstrate proficiency in handling Strings and File Systems.
3. Create, run and manipulate Python Programs using core data structures like Lists,Dictionaries and use Regular Expressions.
4. Interpret the concepts of Web Programming and GUI in Python
5. Apply concepts of Python programming in various fields related to IOT, Web Services and Databases in Python.

UNIT-I

Python basics, Objects- Python Objects, Standard Types, Other Built-in Types, Internal Types, Standard Type Operators, Standard Type Built-in Functions, Categorizing the Standard Types, Unsupported Types

Numbers - Introduction to Numbers, Integers, Floating Point Real Numbers, Complex Numbers, Operators, Built-in Functions, Related Modules

Sequences - Strings, Lists, and Tuples, Dictionaries and Set Types

Control Flow, Truthiness, Sorting, List Comprehensions, Generators and Iterators

UNIT-II

Files: File Objects, File Built-in Function [open()], File Built-in Methods, File Built-in Attributes, Standard Files, Command-line Arguments, File System, File Execution

Exceptions: Exceptions in Python, Detecting and Handling Exceptions, Context Management, Exceptions as Strings, Raising Exceptions, Assertions, Standard Exceptions, Creating Exceptions, Why Exceptions (Now)?, Why Exceptions at All?, Exceptions and the sys Module, Related Modules

Modules: Modules and Files, Namespaces, Importing Modules, Importing Module Attributes, Module Built-in Functions, Packages, Other Features of Modules

UNIT-III

Regular Expressions: Introduction, Special Symbols and Characters, Res and Python**Multithreaded Programming**: Introduction, Threads and Processes, Python, Threads, and the Global Interpreter Lock, Thread Module, Threading Module, Related Modules

UNIT-IV

GUI Programming: Introduction, Tkinter and Python Programming, Brief Tour of Other GUIs, Related Modules and Other GUIs

Web Programming: Introduction, Wed Surfing with Python, Creating Simple Web Clients, Advanced Web Clients, CGI-Helping Servers Process Client Data, Building CGI Application, Advanced CGI, Web (HTTP) Servers

UNIT-V

Database Programming: Introduction, Python Database Application Programmer's Interface (DBAPI), Object Relational Managers (ORMs), Related Modules

Text Book(s)

1. Core Python Programming, Wesley J. Chun, Second Edition, Pearson.
2. Think Python, Allen Downey, Green Tea Press.

Reference Books

1. Introduction to Python, Kenneth A. Lambert, Cengage.
2. Python Programming: A Modern Approach, Vamsi Kurama, Pearson.
3. Learning Python, Mark Lutz, O' Really.

SUGGESTED CO-CURRICULAR ACTIVITIES & EVALUATION METHODS:

Unit 1: Activity: Hands-on Lab exercise on Python Control Statements
Evaluation Method: Lab Performance and Correctness of solution Implementation

Unit 2: Activity: Assignment of Files in Python
Evaluation Method: Problem Solving, Accuracy

Unit 3: Activity: Exercises on Regular expressions
Evaluation Method: Solutions, Accuracy of Validation

Unit 4: Activity: Poster Presentation on various GUI components in Python
Evaluation Method: Content knowledge, organization, clarity, presentation skills, visualaids.

Unit 5: Activity: Group Project
Evaluation Method: Project effectiveness, User interface, Solution to the Problem

V Semester

Course 15 B: Application Development using Python

Credits -1

List of Experiments:

1. Write a menu driven program to convert the given temperature from Fahrenheit to Celsiusand vice versa depending upon user's choice.
2. Write a python program to calculate total marks, percentage and grade of a student. Marks obtained in each of the three subjects are to be input by the user. Assign grades according to the following criteria :

   Grade A: Percentage >=80 Grade B: Percentage>=70 and 80

   Grade C: Percentage>=60 and <70 Grade D: Percentage>=40 and <60 Grade E: Percentage<40

3. Demonstrate various methods of Sequence Data Types
4. Write a python program to display the first n terms of Fibonacci series.
5. Write a python program to calculate the sum and product of two compatible matrices.
6. Write a function that takes a character and returns True if it is a vowel and False otherwise.
7. Write a program to implement exception handling.
8. Write a program to implement Multithreading
9. Develop a Python GUI calculator using Tkinter
10. Write a Python program to read last 5 lines of a file.
11. Design a simple database application that stores the records and retrieve the same
12. Design a database application to search the specified record from the database.
13. Design a database application to that allows the user to add, delete and modify the records.

VII Semester

Course 16 A: Advanced Data Structures

Credits -3

Learning Objective:

To familiarize with the organization of data so as to optimize the searching time

Learning Outcomes: Upon completion of the course, students will be able to:

1. Apply appropriate hashing techniques for a given problem.
2. Simulate the operations of Heap trees.
3. Provide solutions using multi-way search trees.
4. Choose appropriate algorithm while establishing a network.
5. Apply the knowledge of disjoint sets for solving a given problem.

UNIT-I

Hashing – General Idea, Hash Function, Separate Chaining, Hash Tables without linked lists: Linear Probing, Quadratic Probing, Double Hashing, Rehashing, Hash Tables in the Standard Library, Universal Hashing, Extendible Hashing.

UNIT-II

Priority Queues (Heaps) – Model, Simple implementations, Binary Heap: Structure Property, Heap Order Property, Basic Heap Operations: insert, delete, Percolate down, other Heap Operations.

Binomial Queues: Binomial Queue Structure, Binomial Queue Operations, Implementation of Binomial Queue, Priority Queues in the Standard Library.

UNIT-III

Trees – **AVL**: Single Rotation, Double Rotation, B-Trees, B^+ Trees

Multi-way Search Trees – 2-3 Trees: Searching for an element in a 2-3 Tree, inserting a new element in a 2-3 Tree, deleting an element from a 2-3 Tree.

Red-Black Trees – Properties of red-black trees, rotations, insertion, deletion.

UNIT-IV

Graph Algorithms – Elementary Graph Algorithms: Topological sort, Single Source ShortestPath Algorithms: Dijkstra’s, Bellman-Ford, All-Pairs Shortest Paths: Floyd-Warshall’s Algorithm.

UNIT-V

Disjoint Sets – Equivalence relation, Basic Data Structure, Simple Union and Find algorithms, Smart Union and Path compression algorithm.

Text Books:

1. Fundamentals of Computer Algorithms, Ellis Horowitz, SatrajSahani and Rajasekharam, 2nd Edition, 2009, University Press Pvt. Ltd.
2. Advanced Data Structures, Reema Thareja, S. Rama Sree, Oxford University Press, 2018.

Reference Books:

1. Data Structures and Algorithm Analysis in C++, Mark Allen Weiss, 4 th Edition, 2014, Pearson.
2. Introduction to Algorithms, Thomas H Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, 3 rd Edition, 2009, The MIT Press.

SUGGESTED CO-CURRICULAR ACTIVITIES & EVALUATION METHODS:

Unit 1: Activity: Quiz on hashing techniques, covering concepts, algorithms, and applications.

Evaluation Method: Assess students' understanding of hashing techniques through quiz scores and performance.

Unit 2: Activity: Seminar on Heap Trees

Evaluation Method: Evaluate the clarity, depth of understanding, and presentation skills demonstrated in the seminar.

Unit 3: Activity: Group Project to design and implement a multi-way search tree data structure,along with algorithms for insertion, deletion, and searching.

Evaluation Method: Functionality, correctness, and efficiency of the multi-way search tree implementation.

Unit 4: Activity: Role Play to simulate the process of establishing a network, making algorithmic decisions along the way.

Evaluation Method: Students' understanding and application of network establishment algorithms through their decision-making process during the role play.

Unit 5: Activity: Puzzle Challenge that can be solved using disjoint sets, and encourage them toapply their knowledge to find a solution.

Evaluation Method: Assess the correctness and efficiency of students' solutions to the puzzle or problem involving disjoint sets.

VII Semester

Course 16 A: Advanced Data Structures

Credits -1

List of Experiments:

1. Implement Linear probing Hashing Technique.
2. Implement Quadratic probing Hashing Technique.
3. Implement Binary Heap and its operations.
4. Implement AVL Trees and its operations.
5. Implement the operations on B Trees
6. Implement 2-3 Trees and its operations.
7. Implement the operations of Red-Black trees
8. Implement Dijkstra’s shortest path algorithm.
9. Implement Bellman-Ford shortest path algorithm.
10. Implement Floyd-Warshall’s Algorithm.
11. Implement disjoint sets and its operations.
12. Implement Union and Find algorithms

VII Semester

Course 16 B: Artificial Intelligence

Credits -3

Learning Objective:

To provide students with a comprehensive understanding of artificial intelligence (AI) principlesand techniques

Learning Outcomes: Students after successful completion of the course will be able to:

1. Analyze AI problems and search techniques using underlying assumptions and AI techniques.
2. Apply heuristic search techniques for problem-solving and optimization.
3. Understand knowledge representation approaches and apply predicate logic forrepresenting facts and relationships.
4. Utilize rule-based systems for representing knowledge and apply reasoning techniques for problem-solving.
5. Implement symbolic reasoning under uncertainty and augment problem-solving strategieswith non-monotonic reasoning.

UNIT- I

Problems and Search: What is Artificial Intelligence, The AI Problems, and Underlying Assumption, what is an AI Technique?

Problems, Problems Spaces, and Search: Defining the problem as a state space search,production systems, problems characteristics, issues in the design of search programs.

UNIT- II

Heuristic Search Techniques: Generate-and-test, Hill Climbing, Best-First Search, Problem Reduction, Constraint Satisfaction, Means-Ends Analysis

UNIT- III

Knowedge Representation Issues: Representations and Mapping, Approaches to Knowledge Representation, The frame problem. Using Predicate Logic: Representing simple facts in logic, Representing Isa relationships, predicates, Resolution

UNIT- IV

Representing Knowledge using Rules: Procedural Vs Declarative knowledge, Logic Programming, Forward Vs Backward Reasoning, Matching, Control Knowledge

UNIT- V

Symbolic Reasoning under Uncertainty: Introduction to Non-monotonic Reasoning, Logicsfor Non-monotonic Reasoning, Implementation issues, Augmenting a Problem solver, implementation: DFS, BFS.

Statistical Reasoning: Probability and Bayes Theorem, Certainty Factors and Rule-Based Systems, Bayesian Networks, Dempster-**Shafer** Theory.

Text Books:

1. Russell, S., & Norvig, P. Artificial intelligence: a Modern approach. Third Edition. Pearson new international edition. 2014.

Reference Books:

2. Artificial Intelligence, Second Edition, Elaine Rich, Kevin Knight, Tata McGraw-Hill Edition.

SUGGESTED CO-CURRICULAR ACTIVITIES & EVALUATION METHODS:

Unit 1: Activity: Group discussion on real-world AI problems and possible search techniques.

Evaluation Method: Active Participation, Presentation and analysis of group discussion outcomes.

Unit 2: Activity: Problem-solving tasks using heuristic search algorithms.

Evaluation Method: Assessment of problem-solving approach and solution quality.

Unit 3: Activity: Hands-on activity to create knowledge representations using predicate logic.

Evaluation Method: Evaluation of knowledge representation accuracy and logical reasoning.

Unit 4: Activity: Scenario-based problem-solving using rule-based systems.

Evaluation Method: Assessment of problem-solving approach and solutioneffectiveness.

Unit 5: Activity: Simulation activity to implement symbolic reasoning under uncertainty.

Evaluation Method: Evaluation of simulation results and reasoning accuracy.

VII Semester

Course 16 B: Artificial Intelligence

Credits -1

List of Experiments:

1. Write a Program to Implement Breadth First Search
2. Write a Program to Implement Depth First Search
3. Write a Program to Implement Tic-Tac-Toe game.
4. Write a Program to implement 8-Puzzle problem
5. Write a Program to Implement Water-Jug problem
6. Write a Program to Implement Travelling Salesman problem
7. Write a Program to Implement Towers of Hanoi problem
8. Write a Program to implement 8-Queens problem

VII Semester

Course 17 A: Computer Graphics

Credits -3

Learning Objective:

To Develop a comprehensive understanding of computer graphics principles, techniques, and algorithms, and apply them to create visually appealing 2D and 3D graphics.

Learning Outcomes:

Students after successful completion of the course will be able to:

1. Understand computer graphics fundamentals
2. Perform 2D and 3D
3. Apply window-to-view port transformation and perform line and polygon clipping operations.
4. Determine visible surfaces and apply computer graphics algorithms for depth comparison, back-face removal, and rendering.
5. Apply animation principles, work with Flash interface, and gain an introduction to virtual reality.

UNIT-I

Introduction: Advantage of Computer Graphics and Areas of Applications, Hardware and Software for Computer Graphics- Hard Copy, Display Technologies, Random Scan Display System, Video Controller, Random Scan Display Processor, Raster Graphics, Scan Conversion Algorithms (Line, Circle, Ellipse), Area Filling (Rectangle, Ellipse), Clipping (Lines, Circle, Ellipse), Clipping Polygons

UNIT-II

Two dimensional and three-dimensional transformations: 2-Dimensional transformation, 2-D Translation, Rotation, Scaling, Homogeneous Coordinates, Reflection, Shear transform, 3-dimensional transformation, 3-D Translation, Rotation Scaling, Reflection, Shear.

UNIT-III

Clipping: Window to view port transformation, Clipping, line clipping, Cohen —Sutherland line clipping, Polygon clipping, Sutherland and Gary Hodgman polygon clipping algorithm

UNIT-IV

Visible Surface Determination and Computer Graphics algorithm: Image space and object space techniques, Hidden Surface removal—Depth comparison Z-Buffer Algorithm, Back-Face Removal, The Painter's Algorithm, Scan-Line Algorithm, Light and Color and different color models (RGB,CMY, YIQ)

UNIT-V

Animation and Virtual Reality: Basic Principles of Animation and Types of Animation, Introduction to the flash interface: Setting stage dimensions, working with panels, panel layouts, Layers & Views, Shaping Objects – Overview of shapes, Drawing & Modifying Shapes, Bitmap Images & Sounds

Animation -Principles, Frame by frame animation, tweening, masks, Introduction to virtual reality.

Text Books

1. Foley, J. D., A. V. Dam, S. K. Feiner, J. F. Hughes, Computer Graphics Principle and Practices, Addison Wesley Longman, Singapore Pvt. Ltd.,

Reference Books

1. Hearn Donald, M. P. Baker, Computer Graphics, 2E, Prentice Hall of India Private Limited, New Delhi
2. Robert R & Snow D Flash CS4 Professional Bible, Wiley Publishing

SUGGESTED CO-CURRICULAR ACTIVITIES & EVALUATION METHODS:

Unit 1: Activity: Quiz on computer graphics concepts and terminology.

Evaluation Method: Knowledge of computer graphics principles and concepts

Unit 2: Activity: Hands-on lab sessions on 2D and 3D graphics programming.

aluation Method: Practical assignments evaluating the implementation of 2D and 3Dgraphics operations

Unit 3: Activity: Group Project on window-to-view port transformation and clipping algorithms

Evaluation Method: Project effectiveness, Functionality, Solution to the Problem

Unit 4: Activity: Seminar on visible surface determination algorithms and rendering techniques

Evaluation Method: Presentation and demonstration of projects showcasing the application of rendering algorithms and surface removal

Unit 5: Activity: Workshop on animation principles and Flash interface usage, hands-on experience with virtual reality technologies and tools

Evaluation Method: Individual projects demonstrating the application of animation principles, Flash interface usage, and virtual reality

VII Semester

Course 17 A: Computer Graphics

Credits -1

List of Experiments:

1. Implement Brenham’s line drawing algorithm for all types of slopes
2. Implement area filling algorithms
3. Create and rotate a line about a fixed point and origin.
4. Create and rotate a triangle about the origin and a fixed point.
5. Draw a color cube and spin it using OpenGL transformation matrices.
6. Clip a line using Cohen-Sutherland algorithm.
7. Implement polygon clipping algorithm
8. Implement Z-buffer algorithm
9. Implement Painter’s algorithm.
10. Implement tweening

VII Semester
Course 17 B : Design and Analysis of Algorithms
Credits -3

Learning Objectives:

To design, develop and analyze algorithms to provide optimal solutions.

Learning Outcomes: Upon successful completion of the course, students will be able to:

1. Understand the fundamental concepts of algorithm analysis and design techniques.
2. Apply divide and conquer design techniques for solving problems
3. Analyze the performance of given problem using greedy approach.
4. Analyze the given problem and provide the feasible solution using dynamic programming.
5. Analyze the complexity of a given problem.

UNIT-I

Introduction: Notion of Algorithm, Fundamentals of Algorithmic Problem Solving.

Fundamentals of the Analysis of Algorithm Efficiency: Analysis framework and Asymptotic Notations and Basic Efficiency Classes, Amortized Analysis. Introduction to Brute Force Technique, Exhaustive Search.

UNIT-II

Divide and Conquer: Introduction, Merge sort, Quick sort, Binary Search, Finding Maximum and Minimum, Strassen’s Matrix Multiplication

UNIT-III

The Greedy Method: Introduction, Huffman Trees and codes, Minimum Coin Change problem, Knapsack problem, Job sequencing with deadlines, Minimum Cost Spanning Trees, Single Source Shortest paths.

UNIT-IV

Dynamic Programming: Introduction, 0/1 Knapsack problem, All pairs shortest paths, Optimal Binary search trees, Travelling salesman problem.

UNIT-V

Back Tracking: Introduction, n-Queens problem, Sum of subsets, Hamiltonian cycle.

Branch and Bound: Introduction, Assignment problem, Travelling Salesman problem. **Introduction to Complexity classes:** P and NP Problems, NP Complete Problems.

Text Books:

1. Fundamentals of computer algorithms, Ellis Horowitz, Sartaj Sahni, S. Rajasekharan, Second Edition, 2008, Universities Press.

Reference Books:

1. Introduction to the Design & Analysis of Algorithms, Anany Levitin, Third Edition, 2011, Pearson Education.
2. Data Structures and Algorithm Analysis in C, Mark Allen Weiss, 2002, Pearson.

SUGGESTED CO-CURRICULAR ACTIVITIES & EVALUATION METHODS:

Unit 1: Activity: Algorithm Design Contest.

Evaluation Method: Written exam, assessing understanding and application of algorithmic concepts

Unit 2: Activity: Seminar on Divide and Conquer Problem & Solutions.

Evaluation Method: Presentation, Concept Depth, Suitable Applications in real world domain

Unit 3: Activity: Greedy Algorithm Simulation

Evaluation Method: Simulation exercise, evaluating problem analysis and greedyapproach

Unit 4: Activity: Algorithm Visualization

Evaluation Method: Visual representation of algorithms, understanding, presentation and communication skills

Unit 5: Activity: Quiz on complexity analysis concepts

Evaluation Method: Understanding the Complexity classes and problem Analysis

VII Semester

Course 17 B : Design and Analysis of Algorithms using Java / Python C

Credits -1

List of Experiments:

1. Write a program to implement Merge Sort and analyze its performance.
2. Write a program to implement Quick Sort and analyze its performance.
3. Write a program to find the minimum and maximum in a list of elements and analyze it's performance.
4. Write a program to implement Minimum Cost Spanning Trees and analyze its performance.
5. Write a program to implement Single source shortest path algorithm and analyze its performance.
6. Write a program to implement All pairs shortest path algorithm and analyze its performance.
7. Write a program to implement 0/1 knapscak problem and analyze its performance.
8. Write a program to implement n-Queens problem and analyze its performance.
9. Write a program to implement sum of subsets problem and analyze its performance.
10. Write a program to implement Travelling Sales man problem using Branch and Bound approach and analyze its performance.

VII Semester

Course 18 A: Principles of Machine Learning

Credits -3

Learning Objectives:

To design, develop and analyze algorithms to provide optimal solutions.

Learning Outcomes: Upon successful completion of the course, students will be able to:

1. Understand the features of machine learning to apply on real world problems.
2. Characterize the machine learning algorithms as supervised learning and unsupervised learning, apply and analyze the various algorithms of supervised and unsupervised learning.
3. Analyze the concept of neural networks for learning linear and non-linear activation functions.
4. Identify an appropriate clustering technique to solve real world problems.
5. Choose a suitable machine learning model, implement and examine the performance of the chosen model for a given real world problems.

UNIT-I:

Introduction: What is Machine Learning, Examples of Various Learning Paradigms, Perspectivesand Issues, Version Spaces, Finite and Infinite Hypothesis Spaces, PAC Learning

UNIT -II

Learning a Class from Examples, Linear, Non-linear, Multi-class and Multi-label classification, Generalization error bounds: VC Dimension, **Decision Trees:** ID3, Classification and Regression Trees, Regression: Linear Regression, Multiple Linear Regression, Logistic Regression.

UNIT -III

Neural Networks: Introduction, Perceptron, Multilayer Perceptron, Support vector machines: Linear and Non-Linear, Kernel Functions, K-Nearest Neighbors.

UNIT -IV

Introduction to clustering, Hierarchical: AGNES, DIANA, Partitional: K-means clustering, K-Mode Clustering, Self-Organizing Map, Expectation Maximization, Gaussian Mixture Models, Principal components analysis (PCA)

UNIT -V

Machine Learning in Practice Design, Analysis and Evaluation of Machine Learning experiments, Feature selection Mechanisms, other issues: Imbalanced data, missing values, Outliers.

Text Books:

1. Ethem Alpaydin, Introduction to Machine Learning, MIT Press, Prentice Hall of India, Third Edition 2014

Reference Books:

1. Machine learning, Dr. S. Sridhar and M. Vijaya Lakshmi, Oxford University Press, 2021.
2. Tom Mitchell, Machine Learning, McGraw Hill, 3rd Edition,1997.
3. Sergios Theodoridis, Konstantinos Koutroumbas, Pattern Recognition, Academic Press, 4th edition, 2008, ISBN:9781597492720
4. Charu C. Aggarwal, Data Classification Algorithms and Applications, CRC Press, 2014
5. Charu C. Aggarwal, DATA CLUSTERING Algorithms and Applications, CRC Press, 2014

SUGGESTED CO-CURRICULAR ACTIVITIES & EVALUATION METHODS:

Unit 1: Activity: Case Study of real-world applications of Machine Learning

Evaluation Method: Presentation, Concept Depth, Suitable Applications in real worlddomain

Unit 2: Activity: Seminar on Supervised Machine Learning Algorithms

Evaluation Method: Presentation, Concept Depth, Suitable Applications in real world domain

Unit 3: Activity: Neural Network Activation Function Exploration

Evaluation Method: Hands-on activity, evaluating the understanding and analysis of linear and non-linear activation functions

Unit 4: Activity: Case Study on Clustering

Evaluation Method: Analyzing real-world clustering problems, evaluating the ability to identify and apply appropriate clustering techniques for solving real-world problems

Unit 5: Activity: Project work on Machine Learning Models

Evaluation Method: Real-world project implementation, evaluating the ability to chooseand implement a suitable machine learning model for solving real- world problems

VII Semester

Course 18 A: Principles of Machine Learning Lab using Python / R

Credits -1

List of Experiments:

1. Implement Decision Tree learning.
2. Implement Logistic Regression.
3. Implement classification using Multilayer perceptron.
4. Implement classification using SVM
5. Implement K-means Clustering to Find Natural Patterns in Data.
6. Implement K-mode Clustering
7. Implement Hierarchical clustering.
8. Implement Principal Component Analysis for Dimensionality Reduction.
9. Implement Multiple Correspondence Analysis for Dimensionality Reduction.
10. Implement Gaussian Mixture Model Using the Expectation Maximization
11. Implement k-nearest neighbors' algorithm to classify the iris data set. Print both correctand wrong predictions. Java/Python ML library classes can be used for this problem.
12. Implement the non-parametric Locally Weighted Regression algorithm in order to fit data points. Select appropriate data set for your experiment and draw graphs.

VII Semester

Course 18 B: Software Testing

Credits -3

Learning Objectives:

To provide students with a comprehensive understanding of software testing principles, methodologies, and tools, enabling them to effectively design and execute various levels of testing, automate testing processes using Selenium and automation frameworks.

Learning Outcomes: Upon successful completion of the course, students will be able to:

1. Understand software testing principles and apply effective test case design strategies.
2. Implement and execute different levels of testing
3. Utilize Selenium for automation testing, including handling web elements and utilizing advanced features.
4. Implement and leverage automation testing frameworks for efficient test automation.
5. Apply TestNG framework for advanced test execution, management, and parallel processing.

UNIT-I

Fundamentals: Software Testing Principals – Tester Role in Software Development Manual Testing and Automation Testing

Introduction to testing design strategies: Test case design strategies – Using black box approach to test case design – Random testing – Equivalence class partitioning – Boundary value analysis –Using white box approach to test design – Test adequacy criteria – Coverage and control flow graphs – Covering code logic – Paths – Their role in white box-based test design

UNIT-II

Levels of Testing: The need for levels of testing – Unit test – Unit test planning – Designing the unit tests – The class as a testable unit – The test harness – Running the unit tests and recording results – Integration tests – Designing integration tests – Integration test planning – System test –The different types – Regression testing – Alpha, beta and acceptance tests

UNIT-III

Selenium Basics: Automation Testing, Introduction to Selenium and its Components, Selenium IDE Features, Selenium Download and Installation, Creating Scripts using Firebug and Its Installation, Locator Types

Selenium WebDriver: Selenium WebDriver Installation with Eclipse, Handling Dropdowns, Explicit and Implicit Wait, Handling Alerts/Pop-ups, Handling Web Tables, Frames, Dynamic Elements, Robot API, AutoIT

UNIT-IV

Selenium Framework: Test Automation Framework: Introduction, Benefits of Automation Framework, Types of Automation framework

UNIT-V

Introduction to TestNG: TestNG Framework, TestNG installation, TestNG Annotations and Listeners, TestNG Example, TestNG Process Execution: Batch, Controlled Batch & Parallel

Text Books:

1. Ilene Burnstein, "Practical Software Testing", Springer International Edition, 2003.
2. Srinivasan Desikan and Gopalaswamy Ramesh, "Software Testing – Principles and Practices", Pearson education, 2009.
3. Test Automation using Selenium WebDriver with Java: Step by Step Guide by NavneeshGarg
4. Absolute Beginner Java 4 Selenium Webdriver: Come Learn How to Program for Automation Testing by Rex Allen Jones II

Reference Books:

1. Elfriede Dustin, "Effective Software Testing", Pearson Education.
2. Aditya P. Mathur, "Foundations of Software Testing – Fundamental algorithms and techniques", Dorling Kindersley (India) Pvt. Ltd., Pearson Education

Web Links:

https://www.softwaretestingmaterial.com/types-test-automation-frameworks/
https://www.guru99.com/introduction-to-selenium-grid.html#6

SUGGESTED CO-CURRICULAR ACTIVITIES & EVALUATION METHODS:

Unit 1: Activity: Group discussion on software testing challenges and strategies

Evaluation Method: Assessment of participation and contribution

Unit 2: Activity: Assignment on Creation and execution of unit tests

Evaluation Method: Evaluation of accuracy and coverage of unit tests

Unit 3: Activity: Debugging and troubleshooting of test scripts

Evaluation Method: Assessment of problem-solving skills

Unit 4: Activity: Case Study on Analysis and optimization of automated test execution forefficiency

Evaluation Method: Assessment of performance improvement and resource usage

Unit 5: Activity: TestNG report generation and analysis

Evaluation Method: Assessment of report accuracy and insights

VII Semester

Course 18 B: Software Testing Lab using Selenium

Credits -1

List of Experiments:

1. Study of software testing tools such as Rational Rose Test Suite, Selenium Tool
2. Installation and exploring the Selenium IDE
3. Write a script to open google.com and verify that title is Google and verify that it is redirected to google.co.in
4. Write a script to open google.co.in using chrome browser (ChromeDriver)
5. Write a script to open google.co.in using internet explorer (InternetExplorerDriver)
6. Write a script to create browser instance based on browser name
7. Write a script to search for specified option in the listbox
8. Write a script to print the content of list in sorted order.
9. Write a script to print all the options. For duplicates add entry only once. Use HashSet.
10. Write a script to close all the browsers without using quit() method.
11. Write generic method in selenium to handle all locators and return web element for any locator.
12. Write generic method in selenium to handle all locators containing dynamic wait and returnweb element for any locator.

VII Semester

Skill based Course 19 A: Advanced Java Programming

Credits -3

Learning Objectives:

To provide students with a comprehensive understanding of Java Enterprise Edition (J2EE) and its associated technologies for developing robust and scalable web applications.

Learning Outcomes: Upon successful completion of the course, students will be able to:

1. Understand the multi-tier architecture of J2EE and its implementation in enterprise applications.
2. Develop web applications using Java Servlets and establish database connectivity with JDBC.
3. Create dynamic and interactive web pages using Java Server Pages (JSP) and implementJSP with beans and custom tag libraries.
4. Build enterprise applications using Enterprise Java Beans (EJB) and understand their deployment and configuration.
5. Utilize various Java technologies such as JavaMail, CORBA, and Java RMI for effective communication and distributed computing.

UNIT –I

J2EE Overview & Multi-tier Architecture: Overview of J2SE, J2EE, Advantages of Java, Birth of J2EE, Why J2EE; Distributed Systems, The Tier, J2EE Multi-tier architecture,Implementation of Client-tier, Web-tier, EJB-tier, and EIS-tier, Challenges; J2EE best practices: Enterprise Application Strategy, The Enterprise Application - Client, Session Management, Web-tier and JSPs, EJB-tier, MVC, The Myth of Using Inheritance, Maintainable Classes, Performance Enhancement, Power of Interfaces, Threads, and Notification

UNIT –II

Java Servlets & JDBC: Overview of HTML, XML, and XHTML, Java and XML, Parsing XML,Java Servlets and CGI Programming, A Simple Java Servlet, Anatomy of Servlet, Life Cycle of the Servlet, Deployment Descriptor, Reading data from client, reading HTTP requestheaders, working with cookies, Tracking sessions. Overview of JDBC, JDBC Drivers, JDBC Packages, JDBC Process, Database Connection, Statement, ResultSet, Transaction Processing, Servlet program with JDBC.

UNIT –III

Java Server Pages: Overview of JSP, JSP versus Servlet, JSP Tags: Variables and Objects, Directives, Scripting Elements, Standard Actions, Implicit Objects, Scope, Java Server Pages with Beans, Tomcat, User Sessions, Cookies, Session Objects, JSP with JDBC, Creating Custom JSP Tag Libraries.

UNIT –VI

Enterprise Java Beans: The EJB Container, EJB Classes, EJB Interfaces and Deployment Descriptions: Anatomy, Environment elements, referencing EJB, Sharing resources, Security elements, Query elements, Relationship elements, Assembly elements. Session Java Beans - stateless vs stateful, Entity Java Beans - Container-managed persistence, Bean-managed persistence. Message-driven Beans, JAR, WAR, EAR Files.

UNIT –V

JavaMail, CORBA and RMI: JavaMail API and Java Activation Framework, Protocols, Exceptions, Send Email Message, Retrieving Email Messages, Deleting Email Message. CORBA : The Concept of Object Request Brokerage, Java IDL and CORBA, The IDL Interface. Java RMI: Remote Method Invocation Concept, Server Side, and Client Side

Text Books:

1. Jim Keogh: J2EE : The Complete Reference. Mc Graw Hill
2. H. Schildt: Java 2: The Complete Reference. Mc Graw Hill

Reference Books:

1. Kogent Solutions Inc.: Java Server Programming Java EE 7 (J2EE 1.7), Black Book, Dreamtech Press
2. Subrahmanyam Allaramaju et al.: Professional JSP J2EE 1.3 Edition. Wrox Press
3. K. Qian et al.: Java Web Development Illuminated. Narosa
4. Robert W. Sebesta: Programming the World Wide Web. Pearson

VII Semester

Skill Based Course 19 A: Advanced Java Programming

Credits -1

List of Experiments:

1. Study of software testing tools such as Rational Rose Test Suite, Selenium Tool
2. Write a Java program to retrieve the information from the given URL?
3. Write a java Program to create a servlet to read information from client Registration page
4. Write a java Program to create a JSP page to display a simple message along with currentDate
5. Write a java Program to create a User request page in JSP
6. Write the following (JDBC)
   a. Connect database to Java program
   b. Program to create database table using Java
   c. Program to insert, update, delete & select records
   d. Program to delete record from database
   e. Program to execute batch of SQL statements
   f. Program to execute SQL select query
7. Write the following (EJB)
   a. Create stateless bean component
   b. Create stateless bean client
8. JavaMail Example - Send Mail in Java using SMTP
9. Java RMI - Create and execute the server application program

VII Semester
Skill based Course 19 B: MEAN Stack Development

Credits -3

Learning Objectives:

To provide students with the knowledge and skills necessary to develop web applications using modern web development frameworks and technologies, including JavaScript, Node.js, Express, MongoDB, and AngularJS.

Learning Outcomes: Upon successful completion of the course, students will be able to:

1. Gain a comprehensive understanding of web development frameworks, JavaScript fundamentals, and DOM manipulation.
2. Develop proficiency in creating Node.js applications, handling data I/O operations, and utilizing events and callbacks.
3. Build RESTful services using Node.js and Express framework, mastering HTTP handlingand routing.
4. Acquire knowledge and skills in working with MongoDB, performing CRUD operations,and utilizing Mongoose for database integration.
5. Learn to build single-page applications (SPAs) using AngularJS, implementing two-waydata binding and MVC architecture.

UNIT-I

Basic Web Development Framework, Node.js-to-Angular Stack Components

JavaScript Primer: Defining Variables, Understanding JavaScript Data Types, Operators, Looping, Creating Functions, Variable Scope, JavaScript Objects, Manipulating Strings, Workingwith Arrays, Adding Error Handling, Events and Document Object Model, HandlingJSON data, Understanding JSON Callbacks.

UNIT-II

Learning Node.js: Getting Started with Node.js, Understanding Node.js, Installing Node.js, Working with Node Packages, Concurrency and event loop fundamentals, Creating a Node.js Application, Using Events, Listeners, Timers, and Callbacks in Nodes.js: Node.js Event Model, Adding Work to the Event Queue, Implementing Callbacks.

Handling Data I/O in Node.js: Working with JSON, Using the Buffer Module to Buffer Data, Using the Stream Module to Stream Data, Compressing and Decompressing Data with Zlib

UNIT-III

Understanding HTTP Services in Node.js: Processing URLs, Processing Query Strings and Form Parameters, Understanding Request, Response, and Server Objects. Implement HTTP Clients and Servers in Node.Js

Building REST services using Node JS REST services, Installing Express JS, Express Node project structure, Building REST services with Express framework, Routes, filters, template engines – Jade, ejs.

UNIT-IV

Understanding NoSQL and MongoDB: Why NoSQL? , Understanding MongoDB, MongoDB Data Types, MongoDB Basics and Communication with Node JS Installation, CRUD operations, Sorting, Projection, Aggregation framework, MongoDB indexes, Connecting to MongoDB with Node JS, Introduction to Mongoose, Connecting to MongoDB using mongoose, Defining mongoose schemas, CRUD operations using mongoose.

UNIT-V

Building Single Page Applications with AngularJS Single Page Application – Introduction, Two-way data binding(Dependency Injection), MVC in Angular JS, Controllers, Getting userinput, Loops, Client side routing – Accessing URL data, Various ways to provide data in Angular JS – Services and Factories, Working with filters, Directives and Cookies, The digestloop and use of
$apply.

Text Books:

1. Simon Holmes , “Getting MEAN with Mongo, Express, Angular, and Node”, Second Edition, Manning Publications; 1 edition
2. Node.js, MongoDB and Angular Web Development, Brad Dayley, Brendan Dayley, Caleb Dayley, Pearson Education Inc., 2nd Edition, 2018

Reference Books:

1. Jeff Dickey, “Write Modern Web Apps with Mean Stack”, Peachpit press, 2015
2. Ken Williamson, “Learning Angular JS”, O′Reilly; 1 edition
3. Mithun Satheesh, “Web development with MongoDB and Node JS”, Packt Publishing Limited; 2nd Revised edition.

SUGGESTED CO-CURRICULAR ACTIVITIES

1. Training of students by related industrial experts.
2. Assignments

3. Seminars, Group discussions, Quiz, Debates etc.(on related topics).
4. Building chat application using web socket.
5. Build real time dashboard in MEAN stack using websocket
6. Develop a CURD APP for College Student Database

VII Semester
Skill based Course 19 B: MEAN Stack Development

Credits -1

---

1. Installing the Node.js and its dependencies
2. Creating a Node.js application
3. Implementing http services in Node.js
4. Implementing socket services in Node.js
5. Create registration and login forms with validations using Jscript query
6. Jscript to retrieve student information from student database using database connectivity.
7. Building MongoDB environment and managing collection
8. Manipulating MongoDB documents from Node.js
9. Develop and demonstrate Invoking data using Jscript from Mongo DB.
10. Implementing Express in Node.js
11. Implement the following in Angular JS
    a. Angular JS data binding.
    b. Angular JS directives and Events.
    c. Using angular JS fetching data from MySQL.
12. Understanding Angular and Creating a basic Angular application
13. Create an Online fee payment form using JScript and MongoDB.

VII Semester

Skill based Course 20 A: Mobile Application Development

Credits -3

Learning Objectives:

To provide students with a comprehensive understanding of mobile application development using the Android platform.

Learning Outcomes: Upon successful completion of the course, students will be able to:

1. Gain a solid understanding of mobile application development principles
2. Develop proficiency in setting up the Android development environment
3. Acquire the necessary skills to handle and manage Android resources effectively
4. Develop expertise in designing user interfaces by utilizing a wide range of UI widgets
5. Learn various storage techniques in Android and Understand how to integrate web applications

UNIT-I

Mobile Application Development Introduction, advantages, difference between mobile application, Web application and Hybrid Application.

Android Operating System Introduction, Android Versions with Features, Android Architecture, OHA

UNIT-II

Android Application Development Environment: Introduction of Android Studio, Android SDK, Android Development Tools, Android Virtual Devices, Directory Structure of Android Application, Activity & Application Life Cycle, Anatomy of Android Application, Android Manifest File

UNIT-III

Android Terminologies & Resource handling Terminologies: Context, Activity, Intent, Service, Broadcast Receiver, Fragment

Resources: Working with Different Types of Resources Like String, Dimen, Integer, Drawable, Color, Style, Material Design etc.

Animation: Tween Animation and Frame by Frame Animation

UNIT-IV

UI Widgets: TextView, Button, EditText, CheckBox, RadioButton & RadioGroup, AutoCompleteTextView, Spinner, ImageView, Seekbar, ProgressBar, Dialogs

Android Layouts, Menu and Views Layouts: Linear Layout, Absolute Layout, Frame Layout, Relative Layout, Constraint Layout Creation of Layout Programmatically Menu: Option, Context **Views**: Adapters, ListView, ScrollView, WebView, CardView, RecyclerView

UNIT-V

Android Storage Techniques: Shared Preferences, Files & Directories, SQLite Database Connectivity & Operations, Sharing Data Between Application Using Content Providers.

Web Application Integration Techniques and Android APIs: Introduction of JSON, JSON Parsing, Networking API, Telephony API, Web API, Building and Publishing Application to Online Application Store

Text Books:

1. Lauren Darcey and Shane Conder "Android Wireless Application Development", 2nd Edition, Pearson Education,
2. David Griffiths and Dawn Griffiths, "Head First Android Development: A Brain Friendly Guide", O`Reilly

Reference Books:

1. Mark L Murphy, "Beginning Android", Apress, 2011
2. Prasanna Kumar Dixit, "Android", Vikas Publishing House Pvt Ltd.
3. David Mark, Jack Nutting, Jeff LaMarch, "Beginning iOS 6 Development", Apress

SUGGESTED CO-CURRICULAR ACTIVITIES & EVALUATION METHODS:

Unit 1: Activity: Mobile App Development Workshop

Evaluation Method: Students' understanding through a practical project where they develop a basic mobile application.

Unit II: Activity: Android Studio Setup and Configuration Session

Evaluation Method: Successful installation and configuration of the Android Studio development environment.

Unit III: Activity: Resource Management Challenge

Evaluation Method: Students' ability to efficiently manage and utilize different types of Android resources through a practical exercise or assignment.

Unit IV: Activity: UI Design Competition

Evaluation Method: Creativity, usability, and implementation of UI designs using various UI widgets.

Unit V: Activity: Web Integration Hackathon

Evaluation Method: Functionality, user experience, and successful data sharing between the two components during the hackathon.

VII Semester
Skill based Course 20 A: Mobile Application Development with Android

Credits -1

List of Experiments:

1. Study of various IDEs for Android development
2. Setting up Android Studio in Windows
3. Develop an application that uses GUI components, Font and Colours
4. Develop an application that uses Layout Managers and event listeners.
5. Write an application that draws basic graphical primitives on the screen.
6. Develop an application that makes use of databases.
7. Develop an application that makes use of Notification Manager.
8. Implement an application that uses multi-threading.
9. Develop a native application that uses GPS location information
10. Implement an application that writes data to the SD card.

11. Implement an application that creates an alert upon receiving a message
12. Write a mobile application that makes use of RSS feed
13. Develop a mobile application to send an email.

VII Semester
Skill based Course 20 B: R Programming

Credits -3

Learning Objectives:

To equip students with the knowledge and skills to effectively use R programming language for data analysis, including data manipulation, visualization, and statistical modeling, enabling them to make data-driven decisions and insights.

Learning Outcomes: Upon successful completion of the course, students will be able to:

1. Gain a solid understanding of R programming language
2. Acquire knowledge and skills in manipulating matrices, lists, and data frames, including performing operations and applying functions.
3. Develop the ability to create user-defined functions, handle variable scope, and perform exploratory data analysis, including data preprocessing and descriptive statistics.
4. Learn various data visualization techniques in R, including basic and advanced visualizations, as well as creating 3D plots.
5. Gain proficiency in inferential statistics and regression analysis using R, including simple linear regression and multiple linear regression.

UNIT-I

Introduction to R- Features of R - Environment - R Studio. Basics of R-Assignment - Modes - Operators - special numbers - Logical values - Basic Functions - R help functions - R Data Structures - Control Structures.

Vectors: Definition- Declaration - Generating - Indexing - Naming - Adding &Removing elements
- Operations on Vectors - Recycling - Special Operators - Vectorized if- then else-Vector Equality Functions for vectors - Missing values - NULL values - Filtering & Subsetting.

UNIT-II

Matrices - Creating Matrices - Adding or Removing rows/columns - Reshaping - Operations – Special functions on Matrices.

Lists - Creating List – General List Operations - Special Functions - Recursive Lists.

Data Frames -Creating Data Frames - Naming - Accessing - Adding - Removing - Applying Special functions to Data Frames - Merging Data Frames- Factors and Tables.

UNIT-III

Functions - Creating User-defined functions - Functions on Function Object - Scope of Variables - Accessing Global, Environment -Closures - Recursion.

Input / Output – Reading and Writing datasets in various formats

Exploratory Data Analysis - Data Preprocessing - Descriptive Statistics - Central Tendency - Variability - Mean - Median - Range - Variance - Summary - Handling Missing values and Outliers - Normalization

UNIT-IV

Data Visualization in R: Types of visualizations - packages for visualizations - Basic Visualizations, **Advanced Visualizations and Creating 3D plots.**

UNIT-V

Inferential Statistics with R - Types of Learning - Linear Regression- Simple Linear Regression - Implementation in R - functions on lm() - predict() - plotting and fitting regression line. **Multiple Linear Regression** - Introduction -comparison with simple linear regression -Correlation Matrix - F- Statistic - Target variables Vs Predictors - Identification of significantfeatures - Implementation of Multiple Linear Regression in R.

Text Books:

1. Nina Zumel, John Mount, “Practical Data Science with R”, Manning Publications, 2014.
2. Mark Gardener, “Beginning R - The Statistical Programming Language”, John Wiley & Sons, Inc., 2012.
3. W. N. Venables, D. M. Smith and the R Core Team, “An Introduction to R”, 2013.

Reference Books:

1. Jure Leskovec, Anand Rajaraman, Jeffrey D.Ullman, “Mining of Massive Datasets”,Cambridge University Press, 2014.
2. Nathan Yau, “Visualize This: The Flowing Data Guide to Design, Visualization, and Statistics”, Wiley, 2011.

SUGGESTED CO-CURRICULAR ACTIVITIES & EVALUATION METHODS:

Unit 1: Activity: Self Learning through Online resources

Evaluation Method: Online Quiz to access understanding.

Unit II: Activity: Hands-on Lab Session through Datasets

Evaluation Method: Proficiency in manipulating the Datasets.

Unit III: Activity: Data Analysis Competition

Evaluation Method: Students’ ability to preprocess data, application of DescriptiveStatistics.

Unit IV: Activity: Infographic Presentation on Data Visualization

Evaluation Method: Clarity, effectiveness, and aesthetics of their created visualizations.

Unit V: Activity: Project Work

Evaluation Method: Ability to apply the learnt knowledge.

VII Semester

Skill based Course 20 B: R Programming

Credits -1

List of Experiments:

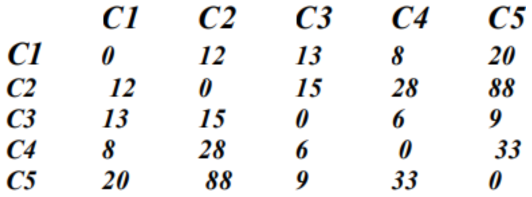
1. Installing R and R studio
2. Installing the "ggplot2", "caTools", "CART" packages and load the packages "ggplot2", "caTools".
3. Basic operations in R
4. Working with Vectors:
    a. Create a vector v1 with elements 1 to 20.
    b. Add 2 to every element of the vector v1.

c. Divide every element in v1 by 5

d. Create a vector v2 with elements from 21 to 30. Now add v1 to v2.

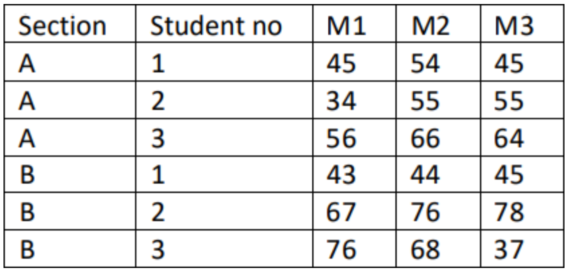
5. Getting data into R, Basic data manipulation

6. Using the data present in the table given below, create a Matrix "M" also Find the pairs of cities with shortest distance.



	C1	C2	C3	C4	C5
C1	0	12	13	8	20
C2	12	0	15	28	88
C3	13	15	0	6	9
C4	8	28	6	0	33
C5	20	88	9	33	0

7. Consider the following marks scored by the 6 students



Section	Student no	M1	M2	M3
A	1	45	54	45
A	2	34	55	55
A	3	56	66	64
B	1	43	44	45
B	2	67	76	78
B	3	76	68	37

a. Create a data structure for the above data and store in proper positions with proper names

b. Display the marks and totals for all students

c. Display the highest total marks in each section.

d. Add a new subject and fill it with marks for 2 sections.

8. Loops and functions - Find the factorial of a given number

9. Implementation of Data Frame and its corresponding operators and functions

10. Implementation of Reading data from the files and writing output back to the specified file

11. Treatment of NAs, outliers, Scaling the data, etc

12. Applying summary() to find the mean, median, standard deviation, etc

13. Implementation of Visualizations - Bar, Histogram, Box, Line, scatter plot, etc.

14. Implementation of Linear and multiple Linear Regression

15. Fitting regression line

VIII Semester

Course 21 A: Big Data Technologies

Credits -3

Learning Objectives:

To provide students with a comprehensive understanding of Big Data technologies, including Apache Hadoop, Hive, HBase, and Zookeeper, and develop practical skills in data processing, querying, and analytics for large-scale datasets.

Learning Outcomes: Upon successful completion of the course, students will be able to:

1. Understand the importance and challenges of Big Data, including its classification and applications.
2. Familiarize with Apache Hadoop and learn data movement and MapReduce algorithms.
3. Explore Hadoop architecture, including HDFS, MapReduce tasks, and cluster setup.
4. Develop skills in Hive and HiveQL for querying and analyzing data in Hadoop.
5. Gain proficiency in HBase, including schema design, advanced indexing, and working with Zookeeper for cluster monitoring.

UNIT- I

INTRODUCTION TO BIG DATA: Introduction – Classification of digital data: Structured, Semi structured and unstructured data, Big Data and its importance, Four V's in Big data, Drivers for Big data, Challenges of Big data, Big data analytics and Big data applications.

UNIT- II

INTRODUCTION HADOOP: Big Data – Apache Hadoop & Hadoop Ecosystem – Moving Data in and out of Hadoop – Understanding inputs and outputs of MapReduce - Algorithms using mapreduce, Matrix-Vector Multiplication by Map Reduce, Data Serialization.

UNIT- III

HADOOP ARCHITECTURE: Hadoop Architecture, Hadoop Storage: HDFS, Common Hadoop Shell commands, Anatomy of File Write and Read., NameNode, Secondary NameNode, and

DataNode, Hadoop MapReduce paradigm, Map and Reduce tasks, Job, TaskTrackers - Cluster Setup – SSH & Hadoop Configuration – HDFS Administering – Monitoring & Maintenance.

UNIT-IV

HIVE AND HIVEQL: Hive Architecture and Installation, Comparison with Traditional Database, HiveQL - Querying Data - Sorting and Aggregating, Map Reduce Scripts, Joins & Subqueries

UNIT-V

HBase concepts- Advanced Usage, Schema Design, Advance Indexing - Zookeeper - how it helps in monitoring a cluster, HBase uses Zookeeper and how to Build Applications with Zookeeper.

Text Books:

1. Big Data Black Book (Covers Hadoop 2, Map Reduce, Hive, Yarn, Pig & Data Visualization) - Dream Tech Publications
2. Big data and Analytics - Seema Acharya and Subhashini Chellappan - Wiley publications.

Reference Books:

1. “Understanding Big data”, Chris Eaton, Dirk deroos et al., McGraw Hill, 2012.
2. “Big Data Analytics”, G. Sudha Sadasivam and R. Thirumahal, Oxford University Press 2020.
3. “HADOOP: The definitive Guide” , Tom White, O Reilly 2012.
4. “Big Data Analytics with R and Haoop”, Vignesh Prajapati, Packet Publishing 2013.
5. “Oracle Big Data Handbook”, Tom Plunkett, Brian Macdonald et al, Oracle Press, 2014.

SUGGESTED CO-CURRICULAR ACTIVITIES:

1. Arrange expert lectures by IT experts working professionally in the area of Big data
2. Assignments
3. Seminars, Group discussions, Quiz, Debates etc.
4. Presentation by students on various applications of Big data.
5. Problem solving exercises.

VIII Semester

Course 21A: Big Data Technologies

Credits -1

List of Experiments:

1. HDFS: Setup a hdfs in a single node to multi node cluster, perform basic file system operation on it using commands provided, monitor cluster performance
2. Write various Map Reduce programs to count the number of times a single word has occurred in a given paragraph.
3. Implement the following file management tasks in Hadoop:
    a. Adding files and directories, List the files and directories
    b. Retrieving files Deleting files
    c. Copying files from one folder to another in HDFS
    d. Copying files from Local File System to HDFS
4. Write a Map Reduce program to add two matrices.
5. Write a Map Reduce program to multiply a matrix with a Vector.
6. Run a basic Word Count Map Reduce program to understand Map Reduce Paradigm
7. Write a Map Reduce program that mines weather data (NCDC). Weather sensors collectingdata every hour at many locations across the globe gather a large volume of log data, whichis a good candidate for analysis with MapReduce, since it is semi structured and record- oriented. (Data available at: ftp://ftp.ncdc.noaa.gov/pub/data/noaa/.)
8. Find average, max and min temperature for each year in NCDC data set
9. Stop word elimination problem:

    Input: 1. A large textual file containing one sentence per line

    2. A small file containing a set of stop words (One stop word per line) **Output**: 1. A textual file containing the same sentences of the large input file without the words appearing in the small file.
10. Write a MapReduce Application to implement Combiners
11. Write a MapReduce Application to implement Reduce-side Join
12. Write a MapReduce Application to implement Map-side Join
13. Hbase: Setup of Hbase in single node and distributed mode, write program to write somedata into hbase and query it

VIII Semester
Course 21 B : Compiler Design
Credits -3

Learning Objectives:

To provide students with a comprehensive understanding of compiler design principles and techniques, including lexical analysis, syntax analysis, intermediate code generation, error handling, storage organization, code generation, and optimization..

Learning Outcomes: Upon successful completion of the course, students will be able to:

1. Understand the compiler structure and the process of lexical analysis using finite automata.
2. Acquire knowledge of syntax analysis techniques, including recursive descent parsing, predictive parsing, and LR parsing.
3. Learn about syntax-directed translation, intermediate code generation, and error detectionand recovery methods in compilers.
4. Explore storage organization, dynamic storage allocation, error recovery methods, andcode generation issues in compilers.
5. Develop an understanding of code optimization techniques, machine-dependent optimization, register allocation, and machine-independent optimization in compilers.

UNIT-I

Overview of the Compiler and its Structure: Language processor, Applications of language processors, Definition-Structure-Working of compiler, the science of building compilers, Difference between interpreter and compiler. Compilation of source code into target language, Types of compilers

Lexical Analysis: The Role of the Lexical Analyzer, Specification of Tokens, Recognition of Tokens, Input Buffering, elementary scanner design and its implementation (Lex), Applying concepts of Finite Automata for recognition of tokens.

UNIT-II

Syntax Analysis: Understanding Parser and CFG (Context Free Grammars), Role of Parser, ParseTree -Elimination of Ambiguity, Left Recursion and Left Factoring of grammar

Syntax Analysis-Top Down: Top Down Parsing - Recursive Descent Parsing - Non Recursive Descent Parsing - Predictive Parsing - LL (1) Grammars.

Syntax Analysis-Bottom Up: Shift Reduce Parsers- Operator Precedence Parsing -LR Parsers, Construction of SLR Parser Tables and Parsing, CLR Parsing, LALR Parsing

UNIT-III

Syntax Directed Definition – Evaluation Order - Applications of Syntax Directed Translation-Syntax Directed Translation Schemes - Implementation of L attributed Syntax Directed Definition. **Intermediate Code Generation:** Variants of Syntax trees - Three Address Code- Types – Declarations - Procedures - Assignment Statements - Translation of Expressions - Control Flow-Back Patching- Switch Case Statements.

UNIT-IV

Error Recovery Error Detection & Recovery, Ad-Hoc and Systematic Methods Source Language Issues, Storage Organization. Stack Allocation of Space, Access to Nonlocal Data on the Stack, Parameter Passing; Symbol Tables; Language Facilities for Dynamic Storage Allocation; Dynamic Storage Allocation Techniques, Heap Management

UNIT-V

Code Generation: Issues in the Design of a Code Generator, the Target Language, Addressesin the Target Code, Basic Blocks and Flow Graphs,

Code Optimization: Optimization of Basic Blocks, A Simple Code Generator, Machine dependent optimization, Register Allocation and Assignment; The DAG Representation of Basic Blocks; Peephole Optimization; Generating Code from DAGs; Design of specifications for compilers, Machine independent optimization Error detection of recovery

Text Books:

1. A. V. Aho, Monica S. Lam, Ravi Sethi and Jeffrey D. Ullman, Compilers: Principles, techniques, &tools, Second Edition, Pearson Education, 2007.
2. K. D. Cooper and L. Torczon, Engineering a compiler, Morgan Kaufmann, 2nd edition, 2011.

3. Steven S.Muchnick, Advanced Compiler design implementation" Elsevier Science India, 2003.
4. Compiler Design by Muneeswaran, Oxford University Press

Reference Books:

1. Andrew A.Appel , Modern Compiler Implementation in Java, Cambridge University Press; 2ndedition, 2002.
2. Allen Holub, Compiler Design in C, Prentice Hall, 1990
3. TorbengidiusMogensen, Basics of Compiler Design, Springer, 2011.
4. Charles N, Ron K Cytron, Richard J LeBlanc Jr., Crafting a Complier, Pearson Education, 2010.

SUGGESTED CO-CURRICULAR ACTIVITIES:

1. Training of students by related industrial experts.
2. Assignments
3. Seminars, Group discussions, Quiz, Debates etc. (on related topics).
4. Presentation by students on Online Compilers and its Architecture
5. Implement the back end of the compiler which takes the three-address code and produces the 8086 assembly language instructions that can be assembled and run using an 8086 assembler. The target assembly instructions can be simple move, add, sub, jump etc.

VIII Semester

Course 21 B: Compiler Design

Credits -1

List of Experiments:

1. Implementation of a Lexical Analyzer using tools like Flex or Lex to recognize and tokenize input programs.
2. Building a Syntax Analyzer using a parser generator like Bison or YACC to verify the syntactical correctness of the input program.

3. Write a LEX program to recognize valid arithmetic expression. Identifiers in the expression could be only integers and operators could be + and *. Count the identifiers & operators present and print them separately.
4. Write a LEX program to eliminate comment lines in a C program and copy the resulting program into a separate file
5. Write YACC program to recognize all strings for which starts with ‘n’ number of ‘a’s followed by n number of ‘b’s.
6. Write YACC program to recognize valid identifier, operators and keywords in the given text (C program) file.
7. Implementation of calculator using lex and YACC.
8. Write a C Program to develop an operator precedence parser for a given language.
9. Convert the BNF rules into YACC form and write code to generate abstract syntax tree.
10. Construct a recursive descent parser for an expression.
11. Construct a Shift Reduce Parser for a given language.
12. Implement Intermediate code generation for simple expressions

VIII Semester

Course 22 A: Data Mining Concepts and Techniques

Credits -3

Learning Objectives:

To provide students with a thorough understanding of data warehousing and data mining concepts, techniques, and applications.

Learning Outcomes: Upon successful completion of the course, students will be able to:

1. Understand data warehousing concepts, including data warehouse architecture, multidimensional data models, and OLAP operations.
2. Explore the fundamentals of data mining, including its definition, techniques, andapplications in real-world scenarios.
3. Develop knowledge and skills in clustering techniques, including partitioning algorithms, hierarchical clustering, and categorical clustering.
4. Acquire proficiency in decision tree construction and the use of decision tree algorithmsfor data analysis and prediction.
5. Gain exposure to various advanced data mining techniques, such as neural networks, genetic algorithms, and text mining, including web mining concepts and applications.

UNIT - I

Data Warehousing: Introduction, What is Data Warehouse? Definition, Multidimensional Data Model, **OLAP** Operations, Warehouse Schema, Data Warehouse Architecture, Warehouse Server, Metadata, OLAP Engine, Data Warehouse Backend Process, Other Features

Data Pre-processing, Descriptive Data Summarization, Data Cleaning, Data Integration and Transformation, Data Reduction, Data Discretization and Concept Hierarchy Generation

UNIT - II

Data Mining: What is Data Mining? Data Mining: Definitions, KDD vs Data Mining, DBMS vs DM, Other Related Areas, DM Techniques, Other Mining Techniques, Issues and Challenges in DM, DM Applications- Case Studies

Association Rules: What is an Association Rule? Methods to Discover Association Rules, A Priori Algorithm, Partition Algorithm, Pincer-Search Algorithm, Dynamic Itemset Counting Algorithms,FP-Tree Growth Algorithm, Discussion on Different Algorithms, Incremental Algorithms, Border Algorithms, Generalized Association Rule, Association Rules with Item Constraints

UNIT - III

Clustering Techniques: Clustering Paradigms, Partitioning Algorithms, k-Medoid Algorithms, CLARA, CLARANS, Hierarchical Clustering, DBSCAN, BIRCH, CURE, Categorical Clustering Algorithms, STIRR, ROCK, CACTUS

UNIT – IV

Decision Trees: What is a Decision Tree? Tree Construction Principle, Best Split, Splitting Indices, Splitting Criteria, Decision Tree Construction Algorithms, CART, ID3, C4.5, Decision Tree Construction with Presorting, Rainforest, Approximate Methods, CLOUDS, BOAT, Pruning Techniques, Integration of Pruning and Construction, Ideal Algorithm

UNIT – V

Other Techniques: What is a Neural Network? Learning in NN, Unsupervised Leaning, Data Mining Using NN: A Case Study, Genetic Algorithms, Rough Sets, Support Vector Machines **Web Mining:** Web Mining, Web Content Mining, Web Structure Mining, Web Usage Mining, Text Mining, Unstructured Text, Episode Rule Discovery for Texts, Hierarchy of Categories, Text Clustering

Text Books:

1. Data Mining Techniques, Arun K Pujari, University Press
2. Data Mining: Concepts and Techniques, 3rd Edition, Jiawei Han, Micheline Kamber, JianPei

SUGGESTED CO-CURRICULAR ACTIVITIES:

1. Arrange expert lectures by IT experts working professionally in the area of Big data
2. Assignments
3. Seminars, Group discussions, Quiz, Debates etc.
4. Presentation by students on various applications of Data Mining.
5. Problem solving exercises.

VIII Semester

Course 22 A: Data Mining Concepts and Techniques

Credits -1

List of Experiments:

1. Study of various Open-Source Data Mining Tools
2. Build Data Warehouse and Explore WEKA
3. Perform data preprocessing tasks and Demonstrate
4. Perform association rule mining on data sets
5. Demonstrate performing classification on data sets
6. Demonstrate performing clustering on data sets
7. Demonstrate performing Regression on data sets
8. Credit Risk Assessment. Sample Programs using German Credit Data
9. Sample Programs using Hospital Management System

VIII Semester
Course 22 B : Digital Image Processing
Credits -3

Learning Objectives:

To provide students with a comprehensive understanding of digital image processing concepts, techniques, and applications.

Learning Outcomes: Upon successful completion of the course, students will be able to:

1. Understand digital image processing fundamentals and applications in various domains.
2. Develop skills in spatial domain image enhancement techniques
3. Acquire proficiency in frequency domain image enhancement
4. Master in image segmentation techniques
5. Learn image compression principles.

UNIT-I

Introduction: Fundamental Steps in Digital Image Processing, Components of an Image Processing System, Sampling and Quantization, Representing Digital Images (Data structure), Some Basic Relationships between Pixels- Neighbors and Connectivity of pixels in image, Applications of Image Processing: Medical imaging, Robot vision, Character recognition, RemoteSensing.

UNIT -II

Image Enhancement in The Spatial Domain: Some Basic Gray Level Transformations, Histogram Processing, Enhancement Using Arithmetic/Logic Operations, Basics of Spatial Filtering, Smoothing Spatial Filters, Sharpening Spatial Filters, Combining Spatial EnhancementMethods.

UNIT -III

Image Enhancement in Frequency Domain: Introduction, Fourier Transform, Discrete Fourier Transform (DFT), properties of DFT, Discrete Cosine Transform (DCT), Image filtering in frequency domain.

UNIT -IV

Image Segmentation: Introduction, Detection of isolated points, line detection, Edge detection, Edge linking, Region based segmentation- Region growing, split and merge technique, local processing, regional processing, Hough transform, Segmentation using Threshold.

UNIT -V

Image Compression: Introduction, coding Redundancy, Inter-pixel redundancy, image compression model, Lossy and Lossless compression, Huffman Coding, Arithmetic Coding, LZW coding, Transform Coding, Sub-image size selection, blocking, DCT implementation using FFT, Run length coding.

Text Books:

1. R. C. Gonzalez and R. E. Woods, Digital Image Processing, 3rd edition, Prentice Hall, 2008.
2. Jayaraman, S. Esakkirajan, and T. Veerakumar, " Digital Image Processing", TataMcGraw-Hill Education, 2011.

Reference Books:

1. Anil K.Jain, "Fundamentals of Digital Image Processing", Prentice Hall of India, 9th Edition, Indian Reprint, 2002.
2. B.Chanda, D.Dutta Majumder, "Digital Image Processing and Analysis", PHI, 2009.

SUGGESTED CO-CURRICULAR ACTIVITIES:

1. Arrange expert lectures in the area of Image Processing.
2. Assignments related to medical image processing, character recognition, signature recognition, remote sensing image processing, etc.
3. Seminars, Group discussions, Quiz, Debates etc. (on related topics).
4. Presentation by students on recent trends of Image processing.

VIII Semester

Course 22 B: Digital Image Processing

Credits -1

List of Experiments:

1. Simulation and Display of an Image, Negative of an Image (Binary & Gray Scale)
2. Implementation of Relationships between Pixels.
3. Implementation of Transformations of an Image
4. Contrast stretching of a low contrast image, Histogram, and Histogram Equalization
5. Display of bit planes of an Image
6. Display of FFT(1-D & 2-D) of an image
7. Computation of Mean, Standard Deviation, Correlation coefficient of the given Image
8. Implementation of Image Smoothening Filters(Mean and Median filtering of an Image)
9. Implementation of image sharpening filters and Edge Detection using Gradient Filters
10. Image Compression by DCT,DPCM, HUFFMAN coding
11. Implementation of image restoring techniques
12. Implementation of Image Intensity slicing technique for image enhancement
13. Canny edge detection Algorithm.

VIII Semester
Course 23 A: Information Security and Cryptography

Credits -3

Learning Objectives:

To provide students with a comprehensive understanding of cryptography and network security concepts and their practical applications.

Learning Outcomes: Upon successful completion of the course, students will be able to:

1. Demonstrate the knowledge of cryptography, network security concepts and applications.
2. Develop security mechanisms to protect computer systems and networks.
3. Apply security principles in system design.
4. Apply methods for authentication, access control, intrusion detection and prevention.
5. Ability to identify and investigate vulnerabilities and security threats and mechanisms to counter them.

UNIT-I

Information Security: Introduction, History of Information security, What is Security, CIA Traid, CNSS Security Model, Components of Information System, Balancing Information Security and Access, Approaches to Information Security Implementation, The Security Systems DevelopmentLife Cycle.

Security Attacks (Interruption, Interception, Modification and Fabrication), Vulnerability, Security Services (Confidentiality, Authentication, Integrity, Non-repudiation, access Control andAvailability) and Mechanisms.

UNIT-II

Cryptography: Concepts and Techniques, Conventional substitution and transposition ciphers, One-time Pad, Block cipher and Stream Cipher, Symmetric and Asymmetric key cryptography, Steganography

Symmetric key Ciphers: DES structure, DES Analysis, Security of DES, variants of DES, Block cipher modes of operation, AES structure, Analysis of AES, Key distribution.

UNIT-III

Asymmetric key Ciphers: Principles of public key cryptosystems, RSA algorithm, Analysis of RSA, Diffie-Hellman Key exchange, Elliptic Curve Cryptography.

Message authentication and Hash Functions, Authentication Requirements and Functions, Message Authentication, Hash Functions and MACs Hash and MAC Algorithms SHA-512, HMAC. Digital Signatures, Authentication Protocols, Digital signature Standard.

UNIT-IV

Program Security: Secure programs, Non-malicious Program errors, Malicious codes virus, Trap doors, Salami attacks, Covert channels, Control against program.

IP Security: Overview, IP Security Architecture, Authentication Header, Encapsulating Security Payload, Combining Security Associations and Key Management.

Email Security: Pretty Good Privacy (PGP) and S/MIME.

UNIT-V

Web Security: Web Security Requirements, Secure Socket Layer (SSL) and Transport LayerSecurity (TLS), Secure Electronic Transaction (SET).

Intruders, Virus and Firewalls: Intruders, Intrusion detection, password management, Virus and related threats, Countermeasures, Firewall design principles, Types of firewalls

Wireless Security, Honeypots, Traffic flow security.

Text Book(s)

1. **Principles of Information Security**: *Michael E. Whitman, Herbert J. Mattord*, CENGAGE Learning, 4th Edition.
2. **Cryptography And Network Security Principles And Practice**, Fourth or Fifth Edition, *William Stallings*, Pearson
3. **Security in Computing**, Fourth Edition, by *Charles P. Pfleeger*, Pearson Education

Reference Books

1. **Modern Cryptography**: **Theory and Practice**, by *Wenbo Mao*, Prentice Hall.
2. **Network Security Essentials**: **Applications and Standards**, by *William Stallings*.Prentice Hall.
3. **Principles of Information Security**, *Whitman*, Thomson.
4. **Cryptography and Network Security** : *Forouzan Mukhopadhyay*, Mc Graw Hill, 2nd Edition

SUGGESTED CO-CURRICULAR ACTIVITIES:

1. Training of students by related industrial experts.
2. Assignments
3. Seminars, Group discussions, Quiz, Debates etc. (on related topics).
4. Presentation by students on emerging Cyber frauds
5. Case Studies of Various Cryptographic Algorithms

VIII Semester

Course 23 A: Information Security and Cryptography

Credits -1

List of Experiments:

1. Write a Java Program to implement Ceaser Cipher
2. Write a Java Program to implement Playfair Cipher
3. Write a Java Program to implement Railfence Cipher
4. Write a Java Program to implement Hill Cipher with 2 x 2 Matrix
5. Write a Java Program to implement DES algorithm
6. Write a Java Program to implement RSA algorithm
7. Write a Java Program for Diffie-Hellman Key Exchange
8. Write a Java Program to Generate SHA-512 Hash of a file
9. Write a Java Program to implement Digital Signature with a File
10. Configuring S/MIME for email communication
11. Setup a honeypot and monitor the honeypot on the network
12. Demonstrate how to provide secure data storage, secure data transmission and for creating digital signatures (GnuPG)
13. Perform wireless audit on an access point or a router and decrypt WEP and WPA (Net Stumbler)
14. Demonstrate intrusion detection system (ids) using any tool (snort or any other s/w)

VIII Semester

Course 23 B: Mobile Ad hoc and Sensor Networks

Credits -3

Learning Objectives:

To provide students with a comprehensive understanding of ad hoc wireless networks, including their fundamentals, protocols, and security mechanisms.

Learning Outcomes: Upon successful completion of the course, students will be able to:

1. Understand the concept of ad-hoc and sensor networks, their applications and typical nodeand network architectures.
2. Describe the MAC protocol issues of ad hoc networks.
3. Identify and describe routing protocols for ad hoc wireless networks with respect to TCP design issues.
4. Explain the concepts of network architecture and MAC layer protocol for WSN.
5. Familiar with the OS used in Wireless Sensor Networks and build basic modules.

UNIT-I

Introduction to Ad Hoc Wireless Networks: Fundamentals of Wireless Communication Technology, The Electromagnetic Spectrum, Radio propagation Mechanisms, Characteristics of the Wireless channel, Cellular and Ad Hoc Wireless Networks, Characteristics of MANETs, Applications of MANETs, Issues and Challenges of MANETs, Ad Hoc Wireless Internet

UNIT-II

MAC Protocols for Ad Hoc Wireless Networks: Introduction, Issues in Designing a MAC protocol for Ad Hoc Wireless Networks, Design goals of a MAC Protocol for Ad Hoc Wireless Networks, Classifications of MAC Protocols, Contention - Based Protocols, Contention - Based Protocols with reservation Mechanisms, Contention – Based MAC Protocols with Scheduling Mechanisms, MAC Protocols that use Directional Antennas, Other MAC Protocols.

UNIT-III

Routing Protocols for Ad Hoc Wireless Networks: Issues in Designing a Routing Protocol, Classifications of Routing Protocols-Table driven protocols- Destination Sequenced Distance Vector (DSDV), Wireless Routing Protocol (WRP), On-demand routing protocol-Dynamic Source Routing

(DSR), Ad Hoc On-Demand Distance Vector Routing (AODV), Hybrid routing protocols-Zone Routing Protocol (ZRP)

UNIT-IV

Transport layer and Security Protocols for Ad hoc Wireless Networks: Introduction, issues in Designing a Transport Layer Protocol for Ad Hoc Wireless Networks. Classification of Transport Layer Solutions. TCP Over Ad Hoc Wireless Networks, Other Transport Layer Protocol for Ad Hoc Wireless Networks.

Security protocols: Security in Ad hoc Wireless Networks, Network Security Requirements, Issues and Challenges in Security Provisioning, Network Security Attacks, Key Management, Secure Routing in Ad hoc Wireless Networks, Cooperation in MANETs, Intrusion Detection Systems.

UNIT-V

Basics of Wireless Sensors and Applications: The Mica Mote, Sensing and Communication Range, Design Issues, Energy Consumption, Clustering of Sensors, Applications, Data Retrieval in Sensor Networks-Classification of WSNs, MAC layer, Routing layer, Transport layer, High- level application layer support, Hardware-Components of Sensor Mote, Sensor Network OperatingSystems– TinyOS, CONTIKIOS, Node-level Simulators – NS2 and its extension to sensor networks, COOJA, TOSSIM

Text Book(s)

1. *C. Siva Ram Murthy and B. S. Manoj*, "**Ad Hoc Wireless Networks Architectures and Protocols**", Prentice Hall, PTR, 2004.
2. *Holger Karl, Andreas willig, "***Protocol and Architecture for Wireless Sensor Networks***"*, John Wiley publication, Jan 2006.

Reference Books

1. *Feng Zhao, Leonidas Guibas*, **"Wireless Sensor Networks: an information processing approach"**, Elsevier publication, 2004.
2. *Charles E. Perkins*, "**Ad Hoc Networking**", Addison Wesley, 2000.
3. I.*F. Akyildiz, W. Su, Sankarasubramaniam, E. Cayirci*, **"Wireless sensor networks: a survey , computer networks"**, Elsevier, 2002, 394 - 422.

SUGGESTED CO-CURRICULAR ACTIVITIES:

1. Training of students by related industrial experts.
2. Assignments
3. Seminars, Group discussions, Quiz, Debates etc. (on related topics).
4. Presentation by students on various Network Simulators
5. Case Studies of Various Applications of Ad hoc and Sensor Networks

VIII Semester

Course 23 B: Mobile Ad hoc and Sensor Networks

Credits -1

List of Experiments:

All the experiments should be done on any Network Simulator like NS-2/NS-2/OMNET++/OPNET etc.

1. Study various network simulators used for wireless Ad-Hoc and Sensor Networks.
2. Introduction to TCL scripting: demonstration of one small Wireless network simulation script.
3. Study various trace file formats of network simulators.
4. Implement and compare various MAC layer protocols.
5. Generate TCL script for UDP and CBR traffic in WSN nodes.
6. Generate TCL script for TCP and CBR traffic in WSN nodes.
7. Implement and compare AODV and DSR routing algorithms in MANET for various parameters.
8. Implement DSDV routing algorithms in MANET.
9. Calculate and compare average throughput for various TCP variants.
10. Implement and compare various routing protocols for wireless sensor networks.
11. Study Ethereal / Wireshark software and analyze dump files.

VIII Semester

Skill based Course 24 A: Advanced Database Management Systems

Credits -3

Learning Objectives:

To provide students with a Through theoretical knowledge and practical application of advanced topics in database management systems.,

Learning Outcomes: Upon successful completion of the course, students will be able to:

1. Gain understanding of relational database concepts, functional dependencies, and correctness of FDs.
2. Analyze and apply normalization techniques (3NF, BCNF, 4NF, 5NF)
3. Develop skills in processing joins, grasp materialized vs. pipelined processing
4. Learn principles of correct interleaved execution, locking mechanisms (2PL), handle deadlocks.
5. Acquire knowledge of T/O-based techniques, multi-version approaches

UNIT-I

Formal review of relational database concepts, Functional dependencies, Closure, Correctness ofFDs

UNIT-II

3NF and BCNF, 4NF and 5NF, Decomposition and synthesis approaches, Review of SQL99, Basics of query processing, external sorting, file scans

UNIT-III

Processing of joins, materialized vs. pipelined processing, query transformation rules, DB transactions, ACID properties, interleaved executions, schedules, serializability

UNIT-IV

Correctness of interleaved execution, Locking and management of locks, 2PL, deadlocks, multiple level granularity, Concurrency Control on B+ trees, Optimistic Concurrency Control and the concepts related to Global and Local transactions in Distributed transactions.

UNIT-V

T/O based techniques, Multiversion approaches, Comparison of Concurrency Control methods, dynamic databases, Failure classification, recovery algorithm, XML and relational databases

Text Book(s)

1. R. Ramakrishnan, J. Gehrke, Database Management Systems, McGraw Hill, 2004
2. A. Silberschatz, H. Korth, S. Sudarshan, Database system concepts, 5/e, McGraw Hill, 2008.

Reference Books

3. Hector Garcia-Molina, Jeff Ullman, and Jennifer Widom, “Database Systems: TheComplete Book”, Pearson, 2011.

SUGGESTED CO-CURRICULAR ACTIVITIES:

1. Performance tuning approaches by subject matter experts
2. Assignments
3. Seminars, Group discussions, Quiz, Debates etc.(on related topics).
4. Creating different kinds of indexes in Oracle and MySQL databases and compare the performance
5. Case study on the need for 2PLand transactional controls

VIII Semester

Skill based Course 24A: Advanced Database Management Systems

Credits -1

List of Experiments:

1. Running Basic SQL commands
2. Understanding the use of Intermediate SQL
3. Running Advanced SQL related to data mining (Slicing and Dicing)
4. Creation of ER and EER diagrams for an organization
5. Database Design and Normalization for a given organization

6. Accessing Databases from Programs using JDBC
7. Analyzing query performance using explain plans
8. Creation of indexes for better query performance.
9. Running different query evaluation plans
10. Experimenting on DBMS locks and session management

VIII Semester
Skill based Course 24 B: Cloud Computing
Credits -3

Learning Objectives:

To provide students with a comprehensive understanding of cloud computing concepts, virtualization technologies, and different service models in the context of cloud computing.

The course will explore the origins, components, and essential characteristics of cloud computing, along with the benefits and limitations associated with its adoption

Learning Outcomes: Upon successful completion of the course, students will be able to:

1. Understand the essential characteristics and benefits of cloud computing
2. Gain knowledge of virtualization technologies
3. Explore Microsoft implementation of virtualization and understand different cloud deployment models and their advantages.
4. Learn about Infrastructure as a Service (IaaS) and Platform as a Service (PaaS) models,
5. Explore Software as a Service (SaaS) and its service providers.

UNIT-I

Cloud Computing Overview – Origins of Cloud computing – Cloud components - Essential characteristics – On-demand self-service, Broad network access, Location independent resource pooling, Rapid elasticity, Measured service. **Cloud scenarios** – Benefits: scalability, simplicity, vendors, security. Limitations – Sensitive information - Application development – **Security concerns** - privacy concern with a third party - security level of third party - security benefits Regularity issues: Government policies.

UNIT-II

Virtualization: Virtualization and cloud computing - Need of virtualization – cost, administration,fast deployment, reduce infrastructure cost - limitations

Types of hardware virtualization: Full virtualization - partial virtualization - para virtualization Desktop virtualization: **Software virtualization** – Memory virtualization - Storage virtualization, **Data virtualization** – **Network virtualization**

UNIT-III

Microsoft Implementation: Microsoft Hyper V, VMware features and infrastructure – Virtual Box - Thin client

Cloud deployment model: Public clouds – Private clouds – Community clouds - Hybrid clouds - Advantages of Cloud computing

UNIT-IV

Infrastructure as a Service (IaaS): IaaS service providers – Amazon EC2, GoGrid, Rack Space, Windows Azure infrastructure services – Amazon EC service level agreement – Recent developments – Benefits

Platform as a Service (PaaS): PaaS service providers – Right Scale – Salesforce.com – Force.com – Oracle APEX cloud - Services and Benefits

UNIT-V

Software as a Service (SaaS): SaaS service providers – Google App Engine, Salesforce.com and google platform – Benefits – Operational benefits - Economic benefits – Evaluating SaaS

Text Book(s)

1. Mastering Cloud Computing, Foundations and Application Programming, Raj Kumar Buyya, Christenveccticla, S Tammaraiselvi, TMH

Reference Books

1. Cloud computing a practical approach - Anthony T.Velte , Toby J. Velte Robert Elsenpeter TATA McGraw- Hill , New Delhi - 2010
2. Cloud Computing: Web-Based Applications That Change the Way You Work and Collaborate Online - Michael Miller - Que 2008
3. Cloud Computing, Theory and Practice, Dan C Marinescu, MK Elsevier.
4. Cloud Computing, A Hands on approach, ArshadeepBahga, Vijay Madisetti, University Press
5. AWS, Azure and Saleforceweb tutorials

SUGGESTED CO-CURRICULAR ACTIVITIES:

1. Training of students by Skill Development Centres
2. Hands-on Lab Sessions on Open Public Clouds
3. Assignments, Seminars, Group discussions, Quiz, Debates etc.(on related topics).
4. Case Studies on operations that can be performed on IaaS, PaaS and SaaS providers

VIII Semester

Skill based Course 24 B: Cloud Computing

Credits -1

List of Experiments:

1. Setup virtual machines on a single computer using VMWare and VirtualBox
2. Create a network using multiple virtual machines on a single host using VMware
3. Setup a client server interaction on a single host using VMware
4. Create an AWS account and create an EC2 instance with a C compiler
5. Connect to EC2 instance and run some C programs on EC2 instance
6. Install a web server on an EC2 instance and provide access to it using Security Group rules
7. Create a virtual cloud on EC2 platform
8. Connect to Force.com and create a data entry form using Salesforce APEX
9. Create a new account on Saleforce.com and create leads, quotes and contracts
10. Analyze the services available on Oracle APEX and create sample web applications

VIII Semester

Skill based Course 25 A: Computer Vision

Credits -3

Learning Objectives:

To equip the students with the knowledge and skills to analyze and interpret images, detect and recognize objects, estimate motion, and apply computer vision techniques in various domains such as biometrics, medical image analysis, surveillance, and augmented reality.

Learning Outcomes: Upon successful completion of the course, students will be able to:

1. Understand the fundamental concepts of computer vision and its applications in various domains.
2. Apply color and geometric transforms, edge-detection techniques, filtering, andmathematical operations to analyze images.
3. Comprehend the concept of motion estimation and its applications.
4. Apply shape correspondence, shape matching, principal component analysis, and shape priors for object recognition.
5. Explore various applications of computer vision

UNIT-I

Introduction to Computer Vision: Image Processing, Computer Vision and Computer Graphics, Computer Vision Applications: Document Image Analysis, Biometrics, Object Recognition,Tracking, Medical Image Analysis, Content-Based Image Retrieval, Video Data Processing, Multimedia, Virtual Reality and Augmented Reality

UNIT-II

Image Representation And Analysis: Image representation, Image processing techniques like color and geometric transforms, Edge-detection Techniques, Filtering, Mathematical operations on image and its applications like convolution, filtering

UNIT-III

Motion Estimation: Introduction to motion, Regularization theory, Optical computation, Stereo Vision, Motion estimation, Structure from motion and models

UNIT-IV

Object Recognition: Hough transforms and other simple object recognition methods, Shape correspondence and shape matching, Principal component analysis, Shape priors for recognition **UNIT-V**

Applications: Photo album, Face detection, Face recognition, Eigen faces, Active appearance and3D shape models of faces Application: Surveillance, foreground background separation, particle filters, Chamfer matching, tracking, and occlusion, combining views from multiple cameras, human gait analysis Application: Invehicle vision system: locating roadway, road markings, identifying road signs, locating pedestrians

Text Book(s)

1. Computer Vision - A modern approach, by D. Forsyth and J. Ponce, Prentice Hall
2. Robot Vision,by B. K. P. Horn, McGraw-Hill.
3. E. R. Davies, Computer & Machine Vision, Fourth Edition, Academic Press, 2012

Reference Books

1. Introductory Techniques for 3D Computer Vision, by E. Trucco and A. Verri, Publisher: Prentice Hall.
2. D. H. Ballard, C. M. Brown. Computer Vision. Prentice-Hall, Englewood Cliffs.
3. Richard Szeliski, Computer Vision: Algorithms and Applications (CVAA). Springer, 2010
4. Image Processing, Analysis, and Machine Vision. Sonka, Hlavac, and Boyle. Thomson.
5. Simon J. D. Prince, Computer Vision: Models, Learning, and Inference, Cambridge University Press, 2012

SUGGESTED CO-CURRICULAR ACTIVITIES:

1. Student Seminar on Applications of Computer Vision
2. Hands-on Lab Sessions on Computer Vision Techniques
3. Assignments, Seminars, Group discussions, Quiz, Debates etc.(on related topics).
4. Project Work

VIII Semester

Skill based Course 25 A: Computer Vision with OpenCV

Credits -1

List of Experiments:

1. Import libraries
2. RGB image and resizing
3. Grayscale image
4. Image denoising, Image thresholding, Image gradients
5. Edge detection fourier transform on image
6. Line transform
7. Corner detection
8. Morphological transformation of image, Geometric transformation of image
9. Contours
10. Image pyramids
11. Color space conversion and object tracking
12. Interactive foreground extraction
13. Image segmentation, Image inpainting
14. Template matching
15. Face and eye detection

VIII Semester
Skill based Course 25 B: Digital Forensics

Credits -3

Learning Objectives:

To equip students with the knowledge and skills necessary to effectively handle digital investigations, ensuring the preservation, analysis, and presentation of digital evidence in a legallysound manner.

Learning Outcomes: Upon successful completion of the course, students will be able to:

1. Gain a clear understanding of the fundamentals of digital forensics
2. Develop knowledge and skills in analyzing storage media and file systems
3. Learn about network forensics and acquire practical skills in network packet sniffing, analysis using tools like Wireshark and TCPDUMP
4. Gain expertise in logs and event analysis, data carving
5. Develop proficiency in wireless and web attacks.

UNIT-I

Introduction to Digital Forensic: Definition of Computer Forensics, Cyber Crime, Evolution of Computer Forensics, Objectives of Computer Forensics, Roles of Forensics Investigator, Forensics Readiness, Steps for Forensics

Computer Forensics Investigation Process: Digital Forensics Investigation Process-Assessment Phase, Acquire the Data, Analyze the Data, Report the Investigation

Digital Evidence and First Responder Procedure: Digital Evidence, Digital Evidence Investigation Process. First Responders Toolkit, Issues Facing Computer Forensics, Types of Investigation, Techniques in digital forensics

UNIT-II

Understanding Storage Media and File System: The Booting Process, LINUX Boot Process, Mac OS Boot Sequence, Windows 10 Booting Sequence, File System, Type of File Systems.

Windows Forensics: Introduction to Windows Forensics, Windows Forensics Volatile Information, Windows Forensics Non- Volatile Information, Recovering deleted files and partitions, Windows Forensics Summary.

Usage of Slack space, tools for Disk Imaging, Data Recovery, Vulnerability Assessment Tools, Encase and FTK tools: **FTK Imager:**

Digital Forensics Road map: Static Data Acquisition from windows using FTK Imager, Live Data Acquisition using FTK Imager
Installation of KALI Linux, RAM Dump Analysis using Volatility, Static Data Acquisition from Linux OS

UNIT-III

Recovering Deleted Files and Partitions: Digital Forensics Tools, Overview of EnCase Forensics, Deep Information Gathering Tool: Dmitry Page, Computer Forensics Live Practical byusing Autopsy and FTK Imager

Network Forensics: Introduction to Network Forensics, Network Components and their forensic importance, OSI internet Layers and their Forensic importance, Tools Introduction Wireshark and TCPDUMP, Packet Sniffing and Analysis using Ettercap and Wireshark, Wireshark Packet Analyzer, Packet Capture using TCP DUMP

Website Penetration: WHOIS, nslookup

UNIT-IV

Logs & Event Analysis: Forensic Analysis using AUTOPSY: Linux and Windows, Forensics andLog analysis, Compare and AUDIT Evidences using Hashdeep Page

Data Carving using Bulk Extractor: Kali Linux and Windows, Recovering Evidence from Forensic Images using Foremost

Application Password Cracking: Introduction to Password Cracking, Password Cracking usingJohn the Ripper, Password Cracking using Rainbow Tables, PDF File Analysis, Remote Imagingusing E3 Digital Forensics

UNIT-V

Wireless and Web Attacks: WiFi Packet Capture and Password Cracking using Aircrack ng, Introduction to Web Attacks, Website Copier: HTTRACK, SQL Injection, Site Report Generation:

Netcraft, Vulnerability Analysis: Nikto, Wayback Machine, Image Metadata Extraction using Imago

Email Forensics Investigation: Email Forensics Investigations, **Mobile Device Forensics:** Mobile Forensics

Preparation for Digital Forensic investigation: Investigative reports, expert witness and cyber regulations, Introduction to Report Writing, Forensic Reports & Expert Witness

Text Book(s)

1. **Digital Forensics**, *Dr.Jeetendra Pande, Dr. Ajay Prasad,* Uttarakhand Open University, Haldwan 2016
2. *Nilakshi Jain, Dhananjay Kalbande,* **"Digital Forensic: The fascinating world of Digital Evidences"** Wiley India Pvt Ltd 2017.
3. *Cory Altheide, Harlan Carvey* **"Digital forensics with open source tools**" Syngress Publishing, Inc. 2011.
4. *Chris McNab,* **Network Security Assessment**, By O'Reilly.

Reference Books

1. *Jason Luttgens, Matthew Pepe, Kevin Mandia,* **"Incident Response and computer forensics",**3rd Edition Tata McGraw Hill, 2014.
2. *Clint P Garrison,* **"Digital Forensics for Network, Internet, and Cloud Computing A forensic evidence guide for moving targets and data"** , Syngress Publishing, Inc. 2010

SUGGESTED CO-CURRICULAR ACTIVITIES:

1. Training of students by related industrial experts.
2. Assignments
3. Seminars, Group discussions, Quiz, Debates etc. (on related topics).
4. Case Studies: Vulnerability Assessment of Your College Website

VIII Semester
Skill based Course 25 B: Digital Forensics
Credits -1

List of Experiments:

1. Study of Computer Forensics and different tools used for forensic investigation
2. How to Recover Deleted Files using Forensics Tools
3. Study the steps for hiding and extract any text file behind an image file/ Audio file (Steganography)
4. How to Extract Exchangeable image file format (EXIF) Data from Image Files using Exifreader Software
5. Data Acquisition using FTK imager
6. How to make the forensic image of the hard drive using EnCase Forensics/Autopsy
7. How to Restoring the Evidence Image using EnCase Forensics/Autopsy
8. How to Collect Email Evidence in Victim PC
9. How to Extracting Browser Artifacts
10. How to View Last Activity of Your PC
11. Find Last Connected USB on your system (USB Forensics)
12. Comparison of two Files for forensics investigation by Compare IT software
13. Live Forensics Case Investigation using Autopsy