Andhra Pradesh State Council of Higher Education

Programme: B.Sc., Honours in Information Technology: MAJOR

w.e.f AY 2023-24

COURSE STRUCTURE

Year	Semester	Course	Title	No. Hrs./ Week	No. of 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 in C - (T)	3	3
			Problem Solving in C - (P)	2	1
		4	Mathematics for Computer Science- (T)	3	3
			Mathematics for Computer Science- (P)	2	1
II	III	5	Database Management Systems- (T)	3	3
			Database Management Systems- (P)	2	1
		6	Python Programming- (T)	3	3
			Python Programming- (P)	2	1
		7	Operating Systems- (T)	3	3
			Operating Systems- (P)	2	1
		8	Computer Architecture - (T)	3	3
			Computer Architecture - (P)	2	1
	IV	9	Web Technologies- (T)	3	3
			Web Technologies- (P)	2	1
		10	Object Oriented Programming using Java- (T)	3	3
			Object Oriented Programming using Java- (P)	2	1
		11	Data Structures Using Python- (T)	3	3
			Data Structures Using Python- (P)	2	1
III	V	12	R Programming- (T)	3	3
			R Programming- (P)	2	1
		13	Software Engineering - (T)	3	3
			Software Engineering - (P)	2	1
		14	Computer Networks- (T)	3	3
			Computer Networks- (P)	2	1
		15	Mobile Application Development - (T)	3	3
			Mobile Application Development - (P)	2	1
	VI		Internship/Apprenticeship		
IV	VII	16	Data Mining Concepts & Techniques- (T)	3	3
			Data Mining Concepts & Techniques- (P)	2	1
		17	Machine Learning using R- (T)	3	3
			Machine Learning using R- (P)	2	1
		18	Network Security & Cryptography- (T)	3	3
			Network Security & Cryptography- (P)	2	1

	VIII	19	Data Science- (T)	3	3
			Data Science- (P)	2	1
		20	Internet of Things - (T)	3	3
			Internet of Things - (P)	2	1
		21	Mean Stack Development- (T)	3	3
			Mean Stack Development- (P)	2	1

Skill Enhanced Courses syllabus will be available in due course of time.

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

principles can be used to explain and predict phenomena in different contexts.

5 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

5. University Physics with Modern Physics by Hugh D. Young and Roger A. Freedman

6. Fundamentals of Physics by David Halliday, Robert Resnick, and Jearl Walker

7. Physics for Scientists and Engineers with Modern Physics" by Raymond A. Serway

and John W. Jewett Jr.

8. Physics for Technology and Engineering" by John Bird

9. Chemistry in daily life by Kirpal Singh

10. Chemistry of bio molecules by S. P. Bhutan

11. Fundamentals of Computers by V. Raja Raman

12. 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:

1. 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.

2. 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:

1. Identifying the attributes of network (Topology, service provider, IP address and bandwidth of

2. your college network) and prepare a report covering network architecture.

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

4. 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:

1. 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.

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. 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.

3. 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.

4. 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:

1. 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.

2. 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.

3. 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

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

2. Identify the networking media used for your college network

3. Identify all the networking devices used in your college premises.

II Semester
Course 3: Problem Solving in C

Credits -3

I. LEARNING OUTCOMES: Upon successful completion of the course, a student will be able to:
1. Understand the functionality of a Digital Computer and fundamental constructs of programming. 2. Analyze and develop solutions to a given problem using control statements.
3. Work with arrays and textual information.
4. Understand the concept of functional hierarchical code organization. 5. Gain knowledge on derived data types and file handling.

UNIT I
Introduction to Computer and Programming:Introduction - Block diagram of a computer -Hardware and Software -Generations of Programming Languages – Algorithms - Flowcharts. Introduction to C: Introduction – Structure of C Program – Writing the first C Program – File used in C Program – Compiling and Executing C Programs – Using Comments – Keywords –Identifiers – Basic Data Types in C – Variables – Constants – I/O Statements in C - Operators in C.
UNIT II
Decision Control and Looping Statements: Decision making statements: if, else if, else if ladder, switch statements; Loop Control Statements: while, do-while, for loop; break, continue and goto statements.
UNIT III
Arrays: Introduction – One Dimensional Arrays - Declaration, Initialization and Memory representation; Two Dimensional Arrays - Declaration, Initialization and Memory Representation; Strings: Declaring and Initializing string variables, character and string handling functions.
UNIT IV
Functions: Introduction – Function declaration/ prototype – Function definition – function call – return statement – Categories of functions - Recursion - Parameter Passing techniques - Scope of variables – Storage Classes.
Pointers: Introduction to Pointers – declaring and initializing pointer Variables – accessing values using pointers - Pointer Arithmetic – Dynamic Memory Allocation.
UNIT V
Structures and Unions: Introduction – Structure definition - accessing structure members – Array of Structures - union definition – difference between structures and unions.
Files: Introduction to Files – Using Files in C – Reading Data from Files – Writing Data to Files – Detecting the End-of-file – Accepting Command Line Arguments.

III. REFERENCES:

TEXT BOOKS:

1. E Balagurusamy – Programming in ANSI C – Tata McGraw-Hill publications.
2. Computer fundamentals and programming in C, REEMA THAREJA, OXFORD UNIVERSITY PRESS

REFERENCE BOOKS:

1. Brain W Kernighan and Dennis M Ritchie - The ‘C’ Programming language - Pearson Publications.
2. Ashok N Kamthane: Programming with ANSI and Turbo C, Pearson Edition Publications.

3. YashavantKanetkar - Let Us ‘C’ – BPB Publications.

IV. SUGGESTED CO-CURRICULAR ACTIVITIES:

1. Assignments (in writing and doing forms on the aspects of syllabus content and outside the syllabus content. Shall be individual and challenging)
2. Student seminars (on topics of the syllabus and related aspects (individual activity))

a. Quiz (on topics where the content can be compiled by smaller aspects and data (Individuals or groups as teams))

b. Study projects (by very small groups of students on selected local real-time
problems pertaining to syllabus or related areas. The individual participation
and contribution of students shall be ensured (team activity))

II Semester

Course 3: Problem Solving in C

Credits -1

IV. PROBLEM SOLVING IN C – PRACTICAL

1. Write a program to find the area of circle and triangle.
2. Write a program to find simple and compound interest.
3. Write a program to convert temperature from Celsius to Fahrenheit
4. Write a program to find whether a number is even or odd
5. Write a program to find sum and average of 5 numbers
6. Write a program to check whether the given number is Armstrong or not.
7. Write a program to find the sum of individual digits of a positive integer.
8. Write a program to generate the first n terms of the Fibonacci sequence.
9. Write a program to find both the largest and smallest number in a list of integer values
10. Write a program to calculate factorial of given integer value using recursive functions
11. Write a program for addition of two matrices.
12. Write a program for multiplication of two matrices.
13. Write a program to perform various string operations.
14. Write a program to search an element in a given list of values.
15. Write a C program to write and read data into/from a File.

II Semester

Course 4: Mathematics for Computer Science

Credits -3

I. LEARNING OUTCOMES:

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

1. Apply mathematical logic to solve problems.
2. Understand sets, relations, functions, and discrete structures.
3. Use logical notation to define and reason about fundamental mathematical concepts such as sets, relations, and functions.
4. Formulate problems and solve recurrence relations.
5. Model and solve real-world problems using graphs and trees.

II.SYLLABUS:

UNIT I

Mathematical logic: Introduction, Statements and Notation, Connectives, Normal Forms, Theory of Inference for the Statement Calculus, The Predicate Calculus, Inference Theory of the Predicate Calculus.

UNIT II

Set theory: Introduction, Basic Concepts of Set Theory, Representation of Discrete Structures, Relations and Ordering, Functions. Algebraic Structures: Introduction, Algebraic Systems, Semi groups and Monoids, Groups, Lattices as Partially Ordered Sets, Boolean algebra.

UNIT III

Elementary Combinatorics: Basics of Counting, Combinations and Permutations, Enumeration

of Combinations and Permutations, Enumerating Combinations and Permutations with Repetitions, Enumerating Permutations with Constrained Repetitions, Binomial Coefficients, The Binomial and Multinomial Theorems, The Principle of Inclusion- Exclusion.

UNIT IV

Recurrence Relations: Generating Functions of Sequences, Calculating Coefficients of generating functions, Recurrence relations, Solving recurrence relations by substitution and Generating functions, the method of Characteristic roots, Solutions of Inhomogeneous Recurrence Relations.

UNIT V

Graphs: Basic Concepts, Isomorphisms and Subgraphs, Trees and their Properties, Spanning Trees, Directed Trees, Binary Trees, Planar Graphs, Euler's Formula, Multigraphs and Euler Circuits, Hamiltonian Graphs, Chromatic Numbers, The Four-Color Problem.

III. REFERENCES: TEXT BOOK (S)

1. Discrete Mathematical Structures with Applications to Computer Science, J.P. Tremblay, R.

Manohar, McGraw Hill education (India) Private Limited. (UNITS - I, II)

2. Discrete Mathematics for Computer Scientists & Mathematicians, Joe L. Mott, Abraham Kandel, Theodore P. Baker, Pearson , 2nd ed. (Units - III, IV, V)
3. Discrete Mathematics by R.K. Bisht and H.S.Dhami, Oxford University Press

REFERENCE BOOKS

1. Discrete Mathematics and its Applications, Kenneth H. Rosen, 7th Edition, McGraw Hill education (India) Private Limited.
2. Discrete Mathematics, D.S. Malik & M.K. Sen, Revised edition Cengage Learning.
3. Elements of Discrete Mathematics, C. L. Liu and D. P. Mohapatra, 4th edition, McGraw Hill education (India) Private Limited.

4. Discrete Mathematics with Applications, Thomas Koshy, Elsevier.
5. Discrete and Combinatorial Mathematics, R. P. Grimaldi, Pearson.

IV. SUGGESTED CO-CURRICULAR ACTIVITIES:

1. Assignments
2. Seminars, Group discussions, Quiz, Debates etc.(on related topics).
3. Presentation by students on applications related to Graph Theory

II Semester

Course 4: Mathematics for Computer Science

Credits -1

MATHEMATICS FOR COMPUTER SCIENCE – PRACTICAL

1. Programming illustration of various propositional logic operations
2. Programming illustration of sets and their operations
3. Implementation of Graphs
4. Illustration of Graph operations including BFS and DFS
5. Implementation of Binary Trees
6. Implementation of various operations on Binary Trees
7. Implementation of Spanning Tree algorithms
8. Illustration of Euler circuits and Hamiltonian circuits
9. Illustration of chromatic number and its applications
10. Illustration of algebraic structure

III Semester
Course 5: Database Management Systems

Credits -3

I. 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.

II. SYLLABUS:

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.

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, 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, Functional dependencies and normal forms upto 3rd normal form and BCNF

UNIT IV

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

UNIT V

PL/SQL: Introduction, Structure of PL/SQL program, PL/SQL Data Types, operators used in PL/SQL, variables, declaring variables in PL/SQL, Creating and running a PL/SQL Program, Control Structures: Conditional control statements, Iterative Control statements, Cursors: Types of cursors, Steps to create a Cursor, using cursors in PL/SQL program.

III. REFERENCES:

TEXT BOOKS:

1. Database management Systems, Alexis Leon and Mathews Leon, Vikas Publications 2002
2. Peter Rob, Carlos Coronel, Database Systems Design, Implementation and Management, Seventh Edition, Thomson (2007)
3. SQL, PL/SQL the Programming Language of Oracle, Ivan Bayross, BPB publications

REFERENCE BOOKS:

1. Elimasri / Navathe, Fundamentals of Database Systems, Fifth Edition, Pearson Addison Wesley (2007).
2. Database Principles, Programming, and Performance, P.O'Neil, E.O'Neil, 2nd ed., ELSEVIER.
3. SQL: The Ultimate Beginners Guide by Steve Tale.
4. Database System Concepts by Abraham Silberschatz, Henry Korth, and S. Sudarshan, McGrawhill
5. Database Management Systems by Raghu Ramakrishnan, McGrawhill

IV. SUGGESTED CO-CURRICULAR ACTIVITIES:

1. Assignments (in writing and doing forms on the aspects of syllabus content and outside the syllabus content. Shall be individual and challenging)
2. Student seminars (on topics of the syllabus and related aspects (individual activity))
3. Quiz(on topics where the content can be compiled by smaller aspects and data (Individuals or groups a steams))
4. Study projects (by very small groups of students on selected local real-time problems pertaining to syllabus or related areas. The individual participation and contribution of students shall be ensured team activity)

III Semester
Course 5: Database Management Systems

Credits -1

1. Illustrate the creation of a table with constraints
2. Creation of college database and establish relationships between tables
3. Employee database

An enterprise wishes to maintain a database to automate its operations. Enterprise divided into certain departments and each department consists of employees. The following two tables describes the automation schemas.

Dept (deptno, dname, loc)

Emp (empno, ename, job, mgr, hiredate, sal, comm, deptno)

Generate the following queries using data of above tables.

i. List out all employees details

ii. Display empno, ename, job and sal columns of all employees

iii. Display employee details who are working as „CLERK‟

iv. Find out number of employees working in each department

v. Find out job wise total salaries and number of employees.

vi. Calculate HRA as 30% and DA as 65% of salary

4. Demonstrate the use of GRANT and REVOKE commands to provide authorization

PL/SQL PROGRAMS

5. Write a PL/SQL program to check the given number is Armstrong or not.
6. Write a PL/SQL program to check the given string is palindrome or not.
7. Writ a PL/SQL program to generate multiplication tables
8. Write a PL/SQL code to find the factorial of any number.
9. Write a PL/SQL program to check the given number is palindrome or not.
10. Write a PL/SQL program to display to 10 rows in Emp table based on their job and salary.
11. Write a PL/SQL program to raise the employee salary by 10% for department number 30 people

Write a procedure to update the salary of Employee, who are not getting commission by 10%

III Semester
Course 6: Python Programming
Credits -3

I. LEARNING OUTCOMES:

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

a. Examine Python syntax and semantics and be fluent in the use of Python flow

control and functions.

b. Demonstrate proficiency in handling Strings and File Systems.

c. Create, run and manipulate Python Programs using core data structures like Lists, Dictionaries and use Regular Expressions.

d. Interpret the concepts of Object-Oriented Programming as used in Python.

II. SYLLABUS:

UNIT I

Introduction to Python: Features of Python Language, installing Python, Environment Setup, python syntax, running a python script, Python 2.x Vs Python 3.x, Python Programming basics: Literals, Data Types: Numeric data types: int, float, complex, string data type, python variables, Expressions, comment statements. Operators - Arithmetic operators, Assignment operators, Comparison operators, Logical operators, Identity operators, Membership operators, Bitwise operators

UNIT II

Standard I/O Operations, python casting Control statements- Conditional branching: if-else, nested if, if-elif-else statements, Iterative statements: while loop, for loop, nested loops, pass statement, continue statement, break statement, and else statement used with loops, Programming using Python conditional and loops block

UNIT III

Functions: Introduction, function definition, creating a function, Function Calling, declaration and defining functions, variable scope and lifetime, built-in functions Sequences:Lists: Creating lists, accessing values in lists, list operations, Tuples: Creating Tuples, accessing values in Tuples, Tuple operations. Sets: Creating a set, accessing values in Set, Set operations, and Dictionaries: Creating a dictionary, Accessing values in Dictionary, Dictionary perations

UNIT IV

Strings and Regular expressions: Introduction to strings, String operations, Built-in string methods and functions, comparing strings, Functions in regular expressions. Object Oriented Programming: Classes and Objects, Class method and self arguments, The Init_Method, Class Variables and Object Variables, The _Del_ Method, Public and Private Data Members Private Methods, Built-In Functions to check, Get, Set and Delete class Attributes, Garbage Collection (Destroying Objects).

UNIT V

Inheritance and Polymorphism: Inheriting Classes in Python, Polymorphism and Method overriding, Types of Inheritance, Composition/Containership, Abstract Classes and Interfaces, Exception Handling: Introduction, Handling exceptions, multiple except blocks and multiple exceptions, finally block.

III. REFERENCES:

TEXT BOOKS:

l. "ReemaThareja", Python Programming using problem solving approach, First Edition, Oxford higher Education.

REFERENCE BOOKS:

l . Kenneth A. Lambert, Fundamentals of Python

2. James Payne, Beginning Python using Python 2.6 and Python 3

3. Charles Dierach, Introduction to Computer Science using Python

IV. SUGGESTED CO-CURRICULAR ACTIVITIES:

1. Organize coding competitions where students can solve programming problems using Python

2. Encourage students to create Python-based projects and showcase them in a project exhibition.

3. Encourage students to contribute to open-source Python projects. This activity exposes them to real-world codebases, collaborative development practices, and the Python community.

III Semester
Course 6: Python Programming

Credits -1

V. PYTHON PROGRAMMING - PRACTICAL

1. Write a program to demonstrate different number data types in Python.

2. Write a program to perform different arithmetic operations on numbers in Python.

3. Write a program to create, concatenate and print a string and accessing.

4. Write a Python Script to print the current date in the following format "Sun May 29 02:26:23 IST 2017".

5. Write a program to create, append and remove lists in Python.

6. Write a program to demonstrate working with tuples in Python.

7. Write a program to demonstrate working with dictionaries in Python.

8. Write a Python program to find largest of three numbers.

9. Write a Python program to convert temperatures to and from Celsius, Fahrenheit. [Formula: c/5 = f-32/9]

10. Write a Python program to construct the following pattern, using a nested for loop

```
*
* *
* * *
* * * *
* * * * *
```

11. Write a Python Script that prints prime numbers less than 20.

12. Write a Python program to find a factorial of a number using Recursion.

13. Write a Python program to define a module to find Fibonacci numbers and import the module to another program.

14. Write a Python program to define a module and import a specific function in that module to another program.

15. Write a Python class to convert an integer to a roman numeral.

16. Write a Python class to implement pow(x, n).

17. Write a Python class to reverse a string word by word

III Semester
Course 7: Operating Systems
Credits -3

I. LEARNING 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. Gain a comprehensive understanding of the architecture of the Android operating system.

I. SYLLABUS

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, NonPreemptive and Preemptive Scheduling Algorithms.

UNITIII

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 InterprocessCommunication;ProcessSynchronization,ClassicalProcessSynchronizationProblems:
Producer-Consumer, Reader-Writer.

UNITIV

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

UNITV

File and I/O Management, OS security : Directory Structure, File Operations, File Allocation Methods, Device Management, Pipes, Buffer, Shared Memory, Security Policy Mechanism, Protection, Authentication and Internal Access Authorization Introduction to Android Operating System, Android Development Framework, Android Application Architecture, Android Process Management and File System, Small Application Development using Android Development Framework.

II. REFERENCEBOOKS:

1. Operating System Principles by Abraham Silber Schatz, Peter Baer Galvin and Greg Gagne
(7thEdition) Wiley India Edition.

2. Operating Systems: Internals and Design Principles by Stallings(Pearson)
3. Operating Systems by J.Archer arris(Author), Jyoti Singh (Author)(TMH)
4. Online Resources for UNITV

IV.SUGGESTEDCO-CURRICULARACTIVITIES:

1. Invite professionals or experts in the field of operating systems to deliver guest lectures or workshops.
2. Case studies on notable operating systems.
3. Organize a project where students simulate the behavior of an operating system

III Semester

Course 7: Operating Systems

Credits -1

OPERATING SYSTEMS - PRACTICAL

1. Write program to implement Round Robin CPU Scheduling algorithm
2. Simulate SJFCPU Scheduling algorithm
3. Write a program the FCFS CPU Scheduling algorithm
4. Write a program to Priority CPU Scheduling algorithm
5. Simulate Sequential file allocation strategies
6. Simulate Indexed file allocation strategies
7. Simulate Linked file allocation strategies
8. Simulate VT and Metamemory management techniques
9. Simulate Single level directory File organization techniques
10. Simulate Two level File organization techniques
11. Simulate Hierarchical File organization techniques
12. Write a program for Bankers Algorithm for Dead Lock Avoidance
13. Implement Bankers Algorithm Dead Lock Prevention.
14. Simulate all Pagere placement algorithms.

a) FIFO
b) LRU
c) LFU

15. Simulate Paging Techniques of memory management

III Semester
Course 8: Computer Architecture

Credits -3

I. LEARNING 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.

II. SYLLABUS:

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 data transfer, 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.

III. REFERENCES

TEXT BOOK:

1. Computer Organization – Carl Hamacher, Zvonko G- Vranesic, Safwat G, Zaky Fifth Edition, Mc- Grawhill INC..,

REFERENCES

1.MansafAlam& Bashir Alam: Digital Logic Design. PHI

2. M. Morris Mano: Digital Logic and Computer Design. Pearson

3. M. Morris Mano: Computer System Architecture. Pearson

4. William Stalling: Computer Organization and Architecture. Prentice Hall
5. Rajaraman& T. Radhakrishnan: Computer Organization and Architecture. PHI
6. Donald D. Givone: Digital Principles and Design. McGraw Hill

IV. SUGGESTEDCO-CURRICULARACTIVITIES:

1. Assignments
2. Seminars, Group discussions, Quiz, Debates etc. (on related topics).
3. Presentation by students on applications related to Graph Theory

III Semester

Course 8: Computer Architecture

Credits -1

V.COMPUTER ARCHITECTURE – PRATICAL

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 nonrestoring division algorithm.
9. Write assembly language code for A+B*(C-D) using various instruction formats in MASM

IV Semester
Course 9: Web Technologies

Credits -3

I. LEARNING OUTCOMES:

Students after successful completion of the course will be able to

1. To understand the web architecture and web services.
2. To practice latest web technologies and tools by conducting experiments.
3. To design interactive web pages using HTML and Style sheets.
4. To study the framework and building blocks of Integrated Development Environment.
5. To provide solutions by identifying and formulating IT related problems.

II. SYLLABUS:

UNIT I

Internet Language, Understanding HTML, HTML document structure, Create a Web Page, Publishing HTML Pages, Tags in HTML, title tag, Text Alignment tags, Text Formatting tags, heading tags, horizontal rule tag, paragraph tag, break tag.HTML Lists - Ordered List, Unordered List & Definition List – Using colors – Using Images

UNIT II

Horizontal Rule Tag - HTML Tables – Nested Tables - Hyperlinks: Textual, Graphical Links to sections – Multimedia Objects – Frames – Nested Frames – Forms – Form Controls: textbox, password, checkbox, radio button, select, text area - Processing of forms

UNIT III

JavaScript- Introduction, simple programming, Obtaining User Input with prompt Dialogs, Operators (arithmetic, Decision making, assignment, logical, increment and decrement) Control Structures - if… else selection statement, while, do… while repetitions statement, for statement, switch statement, break and continue statements.

UNIT IV

Functions – program modules in JavaScript, programmer defined functions, function definition, scope rules, global functions, and recursion JavaScript: Arrays, declaring arrays, accessing elements of an array.

UNIT V

Cascading Style Sheets: Introduction – Using Styles: As an attribute, tag & external file –Defining your own styles Properties and values: properties related to Fonts, Backgrounds & colors, text, boxes & borders

Formatting blocks of information: Classes - Divisions – Spans - Layers with suitable examples.

III.REFERENCES:

TEXT BOOK:

1. Internet & World Wide Web - H.M.Deitel, P.J.Deitel, A.B.Goldberg-5th Edition

REFERENCE BOOKS

1. Programming Worldwide Web by RW Sebesta (Pearson)
2. An Introduction to Web Design + Programming by Wang & Katia (Pearson)
3. HTML & XML An Introduction NIIT(PHI)
4. HTML for the WWW with XHTML & CSS by Wlizabeth Castro (Pearson)
5. Fundamentals of the Internet and the World Wide Web by Raymond Green Law And Ellen Hepp (TMH)
6. Internet and Web Technologies by RajKamal(TMH)
7. Internet and WebBasics by NedSnell,BobTemple,TMClark(Pearson)

IV. SUGGESTEDCO-CURRICULARACTIVITIES:

1. Buildawebsitewith10pagesforthecasestudyidentified.
2. Training of students by related industrial experts.
3. Assignments
4. Seminars, Group discussions ,Quiz, Debates etc.(on related topics).
5. Presentation by students on best websites

IV Semester
Course 9: Web Technologies

Credits -1

V.WEB TECHNOLOGIES - PRACTICAL

1. Design web pages for your college containing a description of the courses, departments, faculties, library etc, use href, list tags.
2. Create your class timetable using table tag.
3. Create user Student feedback form (use textbox, text area, checkbox, radio button, select box etc.)
4. Write HTML code to develop a webpage having two frames that divide the webpage into two equal rows and then divide the row into equal columns fill each frame with a different background color.
5. Create your resume using HTML tags also experiment with colors, text , link ,size and also other tags you studied.
6. Design a web page of your home town with an attractive background color, text color, an Image, font etc. (use internal CSS).
7. Use Inline CSS to format your resume that you created.
8. Use External CSS to format your class timetable as you created.
9. Use External, Internal, and Inline CSS to format college web page that you created.
10. Develop a JavaScript to display today's date.
11. Develop simple calculator for addition, subtraction, and multiplication and division operation using JavaScript
12. Create HTML Page with JavaScript which takes Integer number as input and tells whether the number is ODD or EVEN.
13. Create HTML Page that contains form with fields Name, Email, Mobile No, Gender, Favorite Color and a button now write a JavaScript code to combine and display the information in textbox when the button is clicked

IV Semester

Course 10: Object Oriented Programming through Java

Credits -3

III. LEARNINGOUTCOMES:

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
2. Understand the benefits of code reusability achieved through inheritance
3. Demonstrate various classes in different packages and can design own packages
4. Learn the syntax and mechanisms of exception handling in Java
5. Learn how to create and manage threads and establish connections to database using JDBC.

II.SYLLABUS:

UNITI

Introduction to Java: Features of Java, The Java virtual Machine, Structure of Java ProgramNaming Conventions and Data Types: Naming Conventions in Java, Data Types in Java, Literals Operators in Java: Operators

Control Statements in Java: if... else Statement, do... while Statement, while Loop, for Loop, for each loop, switch Statement , break Statement, continue Statement, return Statement Input and Output: Accepting Input from the Keyboard: Reading Input with Scanner and Buffered Reader class, Displaying Output with System. out. Printf (), Displaying Formatted Output with String. format()

UNIT II

Arrays: Types of Arrays, array name. length, Command Line Arguments Strings: Creating Strings, String Class Methods. Introduction to OOPs: Problems in Procedure Oriented Approach, Features of Object- Oriented Programming System (OOPS) Classes and Objects: Object Creation, Initializing the Instance Variables, Access Specifiers, Constructors Methods in Java: Method Header or Method Prototype, Method Body, Understanding Methods, Static Methods, The keyword 'this', Instance Methods.

UNIT III

Inheritance: Inheritance, The keyword 'super', The Protected Specifier, Types of Inheritance Polymorphism: Polymorphism with Variables, Polymorphism using Methods, Polymorphism with Final Methods, final Class Type Casting: Casting Primitive Data Types, Casting Referenced Data Types, The Object Class Abstract Classes: Abstract Method and Abstract Class Interfaces: Interface, Multiple Inheritance using Interfaces Packages: Package, Different Types of Packages, Interfaces in a Package

UNIT – IV

Exception Handling: Errors in Java Program, Exceptions, throws Clause, throw Clause, Types of Exceptions, Re – throwing an Exception Streams: Stream, Creating a File using File Output Stream, Reading Data from a File uing File Input Stream, Creating a File using File Writer, Reading a File using File Reader.

UNIT – V

Threads: Introduction, Thread Life Cycle, Creating a Thread and Running it, Terminating the Thread. Applets: Introduction, Creating an Applet, Uses of Applets, <APPLET> tag, A Simple Applet, Applet Parameters. Java Database Connectivity: Database Servers, Database Clients, JDBC (Java Database Connectivity), Working with Oracle Database, Stages in a JDBC Program,

III.REFERENCES

TEXT BOOKS:

1. CoreJava: An Integrated Approach, Authored by Dr. R. Nageswara Rao & Kogent Learning Solutions Inc.
2. E.Balaguruswamy, Programming with JAVA, A primer, 3e, TATA McGraw-Hill Company
3. John R.Hubbard, Programming with Java, Second Edition, Schaum' soutline Series,TMH.
4. Deitel & Deitel. JavaTM: How to Program, PHI(2007)

IV.SUGGESTEDCO-CURRICULARACTIVITIES:

2. Conduct coding competitions focused on object-oriented programming concepts in Java
3. Provide students with real-world scenarios and ask them to solve the given problems.
4. Assign group projects that require students to work together to create Java programs using OOP concepts

IV Semester
Course 10: Object Oriented Programming through Java

Credits -1

V.OBJECT ORIENTATEDPROGRAMMINGTHROUGH JAVA- PRACTICAL

2. Write a program to read Student Name, Reg.No, Marks [5] and calculate Total, Percentage, Result .Display all the details of students
3. Write a program to perform the following String Operations

a. Read a string
b. Find out whether there is a given sub string or not
c. Compare existing string by another string and display status
d. Replace existing string character with another character
e. Count number of characters in a string

4. JavaprogramtoimplementsAdditionandMultiplicationoftwoNXNmatrices.
5. Java program to demonstrated use of Constructor.
6. Calculate area of the following shapes using method overloading.

   a. Triangle
   b. Rectangle
   c. Circle
   d. Square

7. Implement inheritance between Person (Aadhar, Surname, Name, DOB, and Age)and Student (Admission Number, College, Course, Year)classes where read Data(),display Data()are overriding methods.
8. Java program for implementing Interfaces
9. Java program on Multiple Inheritance.
10. JavaprogramfortodisplaySerialNumberfrom1toNbycreating two Threads
11. Java program to demonstrate the following exception handlings

a. Divided by Zero
b. Array Index OutofBound
c. Arithmetic Exception
d. User Defined Exception

12. Create an Applet to display different shapes such as Circle ,Oval, Rectangle, Square and Triangle.
13. Write a program to create Book (ISBN,Title,Author,Price,Pages,Publisher)tableandperform The following operations

   a. Add book details
   b. Search a book details for a given ISBN and display book details, if available

c. Update a book detail using ISBN

d. Delete book details for a given ISBN and display list of remaining Books

IV Semester
Course 11: Data Structures Using Python

Credits -3

I. LEARNING OUTCOMES: This course will enable students to
1. Implement Object Oriented Programming concepts in Python.
2. Understand Lists, Dictionaries and Regular expressions in Python.
3. Understanding how searching and sorting is performed in Python.
4. Understanding how linear and non-linear data structures works.
5. To learn the fundamentals of writing Python scripts.

II. SYLLABUS:

UNIT I

Object Oriented Programming:Goals, Principles, and Patterns, Software Development, Class Definitions, Inheritance, Namespaces and Object Orientation,Array based Sequences: Python's Sequence Types, Low-Level Arrays, Dynamic Arrays and Amortization

UNIT II

Stacks: The Stack ADT, Simple Array-Based Stack Implementation Queues: The Queue ADT, Array-Based Queue Implementation Double-Ended Queues: The Deque ADT, Implementing a Deque with a Circular Array

UNIT III

Linked Lists Singly Linked Lists - Implementing a Stack with a Singly Linked List, implementing a Queue with a Singly Linked List. Circularly Linked Lists - Implementing a Queue with a Circularly Linked List Doubly Linked Lists - Basic Implementation of a Doubly Linked List

UNIT – IV

Tree: General Trees - Tree Definitions and Properties, The Tree Abstract Data Type; Binary Trees - The Binary Tree Abstract Data Type, Properties of Binary Trees; Implementing Trees - Linked Structure for Binary Trees, Array-Based Representation of a Binary Tree, Linked Structure for General Trees; Tree Traversal Algorithms – Preorder, Inorder and Postorder Traversals, Implementing Tree Traversals in Python

UNIT – V

Graph Algorithms: Graphs, The Graph ADT, Data Structures for Graphs - Edge List Structure,

Adjacency List Structure, Adjacency Map Structure, Adjacency Matrix Structure, Python Implementation of Graph Traversals- Depth-First Search, Breadth-First Search; Minimum Spanning Trees - Prim-Jarn´ık Algorithm, Kruskal's Algorithm

III. REFERENCES

TEXT BOOKS:
1. Data structures and Algorithms in Python, M.T.Goodrich, R.Tomassia and Michael H. Goldwasser, Wiley Student Edition.
2. Data structures and Algorithms using Python, RanceD.Necaise,Wiley Student Edition.

REFERENCE BOOKS:
1. Introduction to Programming in Python, Robert Sedgewick, Kevin Wayne and Robert Dondero, Pearson.
2. Python Programming, SheetalTaneja and Naveen Kumar, Pearson.
3. Exploring Python, Timothy A.Budd, Tata McGraw-Hill Edition.
4. Think Python, Allen B.Downey, O'Reilly,SPD.
5. Python Programming, ReemaThareja, Oxford University Press.

IV. RECOMMENDED CO-CURRICULAR ACTIVITIES:

1. Organize coding competitions on implementing and solving data structure problems using Python

2. Assignments to reinforce the understanding of different data structures and their operations.

3. Individual or group projects that require students to implement specific data structures using Python

Information Technology Major: IV Semester
Course 11: Data Structures Using Python

Practical 02 hours/Week Credits -1

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 a element using a values and index
   d. Delete an existing element
2. Write Programs to implement the Stack operations using an array
3. Write Programs to implement the Stack operations using Linked List.
4. Write Programs to implement the Queue operations using an array.
5. Write Programs to implement the Queue operations using Linked List.
6. Write a program for Binary Search Tree Traversals
7. Write a program to implement dequeuer using a doubly linked list.
8. Write a program to search an item in a given list using Binary Search
9. Write a program for implementation of the Quick sort
10. Write a program for implementation of Insertion sort
11. Write a program to implement DFS graph traversals algorithm
12. Write a program to implement BFS graph traversals algorithm

V Semester

Course 12: R Programming

Credits -3

I. LEARNINGOUTCOMES:

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

1. Gain a solid understanding of R programming language
2. Acquire knowledge on various data structures and control structures in R.
3. Perform vectorized operations in R programming.
4. Develop skills in manipulating and transforming vectors, matrices, arrays and lists in R.
5. Explore and analyze data using data frames and tables.

II. SYLLABUS :

UNITI

Introduction to R: R overview and history, Basic features of R, Benefits of R, data types inR, Installing R, Getting started with the RStudio IDE, Running R, Packages in R, variable names and
assignment, operators, Input/output functions, reading and writing data.

UNIT II

Control structures: Conditional statements, Loops, dates and times, functions, String manipulations. Preview of Some Important R Data Structures:Vectors, Character Strings, Matrices,Lists, Data Frames,and Classes.

UNIT III

Vectors: Scalars,Vectors,Arrays and Matrices: Adding and Deleting Vector Elements, Obtaining the Length of a Vector Common vector operations: Arithmetic & logical operations, Vector Indexing, Generating vector sequences with seq(),Repeating vector constants with rep (), using all () and any () functions, Vectorized operations, NA and NULL values.

UNIT IV

Matrices and Arrays: Creating Matrices, General Matrix operations-linear algebra operations, matrix indexing, filtering on matrices, using apply () function , Add and Delete matrix rows and columns. Lists: Creating Lists, General List Operations, List Indexing Adding and Deleting List Elements, Getting the Size of a List ,Accessing List Components and Values, Using l apply()and apply()functions.

UNIT-V

Data Frames: Creating Data Frames, Accessing Data Frames-Other Matrix-Like Operations: Extracting sub data frames, using r bind()and c bind()functions.

Factors and Tables : Factors and Levels - Common Functions Used with Factors : t apply() ,split() and by() - Working with Tables, Matrix/Array-Like Operations on Tables, Extracting a Sub table-Math Functions: aggregate()and cut()functions.

III. REFERENCES :

TEXTBOOKS:

1. TheArt of R Programming by Norman Matlof, Nostarch press, sanfransisco,2011.
2. An Introduction to R for Beginners by SASHA HAFNER,on AUG-2019

REFERENCEBOOKS:

1. R Programming for Dummies, AndriedeVries and JorisMeys, Wiley
2. R for Data Science, Hadley Wickham, Garrett Grolemund ,O’Reilly Media
3. R Programming: AStep-By-StepGuideforAbsoluteBeginners-2ndEdition,DanielDanielBell
4. LearnRprogrammingin1Day, KrishnaRungta, PublishedbyGuru99

IV.SUGGESTEDCO-CURRICULARACTIVITIES:

1. Assign students real-world data analysis projects that require them to apply their programming skills.
2. Organize coding challenges focused on R Programming.
3. Organize guest lectures or workshops.

V Semester

Course 12: R Programming

Credits -1

V.R PROGRAMMING - PRACTICAL

1) Write an R Program to take in put from user.
2) Write an R Program to demonstrate working with operators (Arithmetic, Relational, Logical, Assignment operators).
3) Write an R Program to Check if a Number is Odd or Even
4) Write an R Program to check if the given Number is a Prime Number
5) Write an R Program to Find the Factorial of a Number
6) Write an R Program to Find the Fibonacci sequence Using Recursive Function
7) Write an R Program to create a Vector and to access elements in a Vector
8) Write an R Program to create a Matrix from a Vector using dim()function.
9) Write an R Program to create a List and modify its components.
10) Write an R Program to create a Data Frame.
11) Write an R Program to access a Data Frame like a List.
12) Write an R Program to create a Factor

V Semester
Course 13: Software Engineering
Credits -3

I. LEARNINGOUTCOMES:

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 practical implementation
4. Analyze and Evaluate Software Maintenance and Evolution Strategies
5. Apply Advanced Object-Oriented Software Engineering Concepts.

II. SYLLABUS:

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 lifecycle(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), Serviceoriented architecture (SOA), Agile software development and Scrum methodologies.

III. REFERENCES:

TEXTBOOK(S)

1. An Introduction to ObjectOrientedAnalysisandDesignandtheUnifiedProcess,3rd Edition, Craig Larman, Prentice-Hall.
2. Programming in Java by Sachin Malhotra, Oxford University Press

REFERENCEBOOKS

1. Requirements engineering: processes and techniques, G.Kotonya and, I.Sommerville,1998, Wiley

2. Design Patterns, E.Gamma, R.Helm,R. Johnson,andJ.Vlissides

3. The Unified Modeling Language Reference Manual, J.Rumbaugh, I.Jacobson and G.Booch,Addison Wesley

IV.SUGGESTEDCO-CURRICULARACTIVITIES:

1. Assign students real-world software development projects that require them to apply software engineering principles and practices.

2. Encourage students to prepare and deliver technical presentations or demonstrations on software engineering topics of their choice.

3. Invite industry professionals and experienced software engineers to deliver guest lectures or conduct workshops

V Semester

Course 13: Software Engineering

Credits -1

V. SOFTWARE ENGINEERING - PRACTICAL

1. To perform the Requirement analysis of the specified problem and draw a flowchart

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: DFDlevel0i.e. Context Diagram and draw it

4. Understanding of System modeling: Functional modeling: DFDlevel1and DFDlevel2and draw it.

5. Understanding different actors and use cases in detail of the specified problem statement and draw it

Using Rational Rose software any other available software.

6. To perform the user’s view analysis: Use case diagram and draw it using Rational Rose or any other available software.

7. To draw the structural view diagram: Class diagram of specified problem statement using Rational Rose or any other available software.

8. To draw the behavioral view diagram: State-chart diagram, Activity diagram of specified problem Statement using Rational Rose any other available software.

9. TounderstandtestingandperformBoundaryvalueanalysisandEquivalenceclasstesting.

10. To draw Flow graph, DD paths , calculation of cyclomatic complexity and find out all the Independent paths from the D D paths graph.

11. Case study: Prepares RS for a given problem statement

V Semester
Course 14: Computer Networks
Credits -3

I. LEARNING OUT COMES:

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.

II. SYLLABUS:

UNITI

INTRODUCTION: Network applications, network hardware, network software, reference models :OSI, TCP/IP, Internet, Connection oriented network X.25, frame relay. THEPHYSICALLAYER: Theoretical basis for communication, guided transmission media, wireless transmission, the public switched telephone networks , mobile telephone system.

UNITII

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.

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

UNIT-III

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

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

UNIT-V

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.

III. REFERENCES

TEXT BOOKS

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

REFERENCEBOOKS

1. Behrouz A. Forouzan (2006), Data communication and Networking, 4th Edition, McGraw Hill,India.
2. Kurose,Ross(2010),ComputerNetworking:Atopdownapproach,PearsonEducation,India.

IV.SUGGESTEDCO-CURRICULARACTIVITIES:

1. Hands-on exercises to configure network applications
2. Guest Lectures and Workshops on routing algorithms, congestion control, and network layer protocols
3. Group Projects on Network Application Development

V Semester

Course 14: Computer Networks

Credits -1

V.COMPUTER NETWORKS – PRACTICAL

1. Introduction to networking tools and Linux
2. Introduction to Packet Tracer tool from Cisco
3. Study different types of network cables
4. Study different types of networks in detail
5. Study the basics of TCP/IP using various networking tools available in Linux
6. Create a network topology using packet tracer
7. Configure routing using packet tracer
8. Study network security algorithms
9. Implements DNS using packet tracer
10. Implement SMTP connectivity

V Semester

Course 15: Mobile Application Development

Credits -3

I.LEARNINGOUTCOMES:

1. Lean-to setup a new Material App using Android Studio.
2. Understand the Widget tree and learn trouser-made Flutter, widgets for user interface design.
3. Learn to incorporate Image and Text Widgets to create simple user interfaces.
4. Learn to customize pre-built Flutter widgets.
5. Adding App Icons for iOS and Android builds.
6. Learn to run Flutter apps on iOS Simulator, Android Emulator and physical iOS and Android devices

II.SYLLABUS:

UNITI

Introduction:FrameworksandToolsforMobileAppDevelopment,CharacteristicsofMobileApplications, History of Mobile Application Frameworks and Tools, Introduction to Android, iOS,andFlutter.Client-ServerArchitecture:1-tier,2-tier,3-tier, types of Connection, Synchronization, Mobile Device Types, Mobile Device Components, Types of Mobile Applications.

UNITII

Mobile Application Development using Flutter: to set up a new Material App using Android Studio, Creating UI with Flutter: Using Hot Reload and Hot Restart to quickly refresh the app UI and understand when to use each, using the Pubspec. yaml file to incorporate, dependencies, custom assets and fonts, an introduction to the Widget build() method, using layout widgets such as Columns, Rows, Containers and Cards, incorporating Material icons using the Icons class.

UNIT-III

Building Apps with State: Understanding the difference between Stateful and Stateless widgets and when they should each be used, understanding how callbacks can be used to detect user interaction in button widgets, declarative style of UI programming and how Flutter widgets react to state changes, importing dart libraries to incorporate additional functionality, variables, data types and functions work in Dart, building flexible layouts using the Flutter Expanded widget, relationship between setState(), State objects and Stateful Widgets.

UNIT-IV

Using the Dart package manager: to use Dart package manager to incorporate Flutter compatible packages into your projects, functions in Dart and the arrow syntax, to refactor widgets and understand Flutter's philosophy of UI as code. Structuring Flutter Apps: to use Dart Constructors to create customisable Flutter widgets, apply common mobile design patterns to structure Flutter apps.

UNIT-V

Security: User to Mobile Client Security Issues, Mobile Client Security Issues, Client-Server Communications Security Issues, Existing Web Architectures and Back-End Systems Security Issues, Mobile Application Development Management

III.REFERENCES

1. Lee,H.Schneider,andR.Schell:MobileApplications:Architecture,Design,andDevelopment.Pearson
2. MarcoL.Napoli:Beginning Flutter: A Hands on Guide to App Development. Wiley
3. BillPhillips&Brian Hardy: Android Programming the Big NerdRanch Guide.BigNerdRanch
4. Brian Fling: Mobile Design and Development. O'Reilly

V Semester

Course 15: Mobile Application Development

Credits -1

IV.MOBILE APPLICATION DEVELOPMENT – PRACTICAL

1. Mobile application Project Ideas

   i) Tic Tac Toe Game

   ii) Online voting system

   iii) Photo Management Application

   iv) Online Exam Application

VII Semester

Course 16: Data Mining Concepts & Techniques

Credits -3

I. 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, and applications 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 algorithms for 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.

II. SYLLABUS

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

III. Text Books:

1. Data Mining Techniques, Arun K Pujari, University Press

2. Data Mining: Concepts and Techniques, 3rd Edition, Jiawei Han, Micheline Kamber, Jian Pei

IV. 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

VII Semester
Course 16: Data Mining Concepts & Techniques

Credits -1

V.DATA MINING CONCEPTS AND TECHNIQUES– PRACTICAL

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

VII Semester

Course 17: Machine Learning using R

Credits -3

I. LEARNING OUTCOMES: After completing this course, the student will be able to

1. Understand the basic concepts such as decision trees and neural networks.
2. Ability to formulate machine learning techniques to respective problems.
3. Apply machine learning algorithms to solve problems of moderate complexity.
4. Use and program in the programming language R

II. SYLLABUS:

UNIT – I

INTRODUCTION: Overview of Machine learning, machine learning types, well posed learning algorithms and issues in machine learning. **CLASSICAL DATA ANALYSIS:** Mean, variance, Regression- linear. **CONCEPT LEARNIN**G: Introduction, a concept learning task and learning as search, version space and candidate elimination algorithm, inductive bias.

UNIT – II

DECISION TREE LEARNING: - introduction, Representation, basic decision tree learning algorithm, space search in decision tree learning algorithm , cross validation and over fitting.

NEURAL NETWORK LEARNING: Introduction, Perceptrons - Representational power of Perceptrons, Back propagation algorithm.

UNIT – III

INTRODUCTION TO R: Basic features of R, Benefits of R, data types in R, Installing R, Getting started with the RStudio IDE, Running R, Packages in R, variable names and assignment, operators, Input/output functions.

UNIT-IV

Preview of Some Important R Data Structures: Vectors, Character Strings, Matrices, Lists, Data Frames, and Classes.

Control Structure: Conditional statements, Loops, dates and times, functions, String manipulations.

UNIT-V

Vectors: Scalars, Vectors, Arrays and Matrices: Adding and Deleting Vector Elements, Obtaining the Length of a Vector, Vector Indexing, Common vector operations, Generating vector sequences, Repeating vector constants, using all() and any(), Vectorized operations.

III. REFERENCES:

TEXT BOOKS:

1. Tom Michel, Machine Learning, McGraw Hill, 1997.
2. The Art of R Programming by Norman Matlof, No starch press, SAN FRANSISCO,2011.

REFERENCE BOOKS:

1. Machine Learning Methods in the Environmental Sciences, Neural Networks, William W Hsieh, Cambridge University Press.
2. An Introduction to R for Beginners by SASHA HAFNER, on AUG-2019.

IV. <u>SUGGESTED CO-CURRICULAR ACTIVITIES:</u>

1. Assignments
2. Student seminars
3. Quiz
4. Study Projects

5. **VII Semester**
6. **Course 17: Machine Learning using R**
7. Credits -1

V. MACHINE LEARNING WITH 'R' - PRACTICALS

1. Write an R-Program to take input from the user.
2. Write an R-Program to compute a product of two integer values.
3. Write an R-Program to demonstrate working with operators (Arithmetic, Relational, Logical, Assignment operators).
4. Write an R Program to Check if a Number is Odd or Even 5. Write an R program to check whether the biggest of two values.
6. Write an R Program to check if the given Number is a Prime Number
7. Write an R Program to Find the Factorial of a Number
8. Write an R Program to Find the Fibonacci sequence Using Recursive Function.
9. Write an R Program to create a Vector and to access elements in a Vector.
10. Write an R Program to create a Matrix from a Vector using dim() function.

VII Semester
Course 18: Network security & Cryptography
Credits -3

I. LEARNING OUTCOMES: Upon successful completion of the course, a student will be able to:

1. Extensive, thorough and significant understanding of the concepts, issues, principles and theories of computer network security
2. Identifying the suitable points for applying security features for network traffic
3. Understanding the various cryptographic algorithms and implementation of the same at software level
4. Understanding the various attacks, security mechanisms and services

II. SYLLABUS:

UNIT- I

Computer Security Concepts, The OSI Security Architecture, Security Attacks, Security Services and Security Mechanisms, Classical Encryption Techniques- Symmetric Cipher Model, Substitution Ciphers, Transposition Ciphers, Steganography, Modern Block Ciphers, Modern Stream Ciphers. Modern Block Ciphers: Block Ciphers Principles, Data Encryption Standard (DES), Linear And Differential Cryptanalysis, Block Cipher Modes Of Operations, AES.

UNIT- II

Public-Key Cryptography: Principles Of Public-Key Cryptography, RSA Algorithm, DiffieHellman Key Exchange, Elgamal Cryptographic System, Elliptic Curve Arithmetic, Elliptic Curve Cryptography Cryptographic Hash Functions: Applications Of Cryptographic Hash Functions, Requirements And Security, Hash Functions Based On Cipher Block Chaining, Secure Hash Algorithm (SHA).

UNIT- III

Message Authentication Codes: Message Authentication Requirements, Message Authentication Functions, Requirements For Message Authentication Codes, Security Of Macs, HMAC, Macs Based On Block Ciphers, Authenticated Encryption. Digital Signatures and Authentication Protocols: Digital Signatures, Authentication Protocols.

UNIT- IV

Key Management and Distribution: Symmetric Key Distribution Using Symmetric Encryption, Symmetric Key Distribution Using Asymmetric, Distribution Of Public Keys,

X.509 Certificates, Public Key Infrastructure. Electronic Mail Security: Pretty Good Privacy (PGP), S/MIME

UNIT-V

Security at The Transport Layer (SSL And TLS): SSL Architecture, Four Protocols, SSL Message Formats, Transport Layer Security, HTTPS, SSH Security At The Network Layer (Ipsec): Two Modes, Two Security Protocols, Security Association, Security Policy, Internet Key Exchange. Intruders: Intruders, Intrusion Detection, Password Management, Firewalls, Viruses and Worms.

III. REFERENCES:

TEXT BOOKS:

1. Cryptography and Network Security: Principles and Practice, William Stallings, Fifth Edition, Pearson Education.
2. Cryptography and Network Security, Behrouz A. Frouzan and Debdeep Mukhopadhyay, 2nd edition, McGraw Hill Education
3. Network Security and Cryptography, Bernard Menezes , Cengage Learning.
4. Cryptography and Security, C.K. Shyamala, N. Harini and Dr. T.R. Padmanabhan, Wiley-India.

REFERENCE BOOKS:

1. Applied Cryptography, Bruce Schiener, 2nd edition, John Wiley & Sons.
2. Cryptography and Network Security, Atul Kahate, TMH.
3. Introduction to Cryptography, Buchmann, Springer.
4. Number Theory in the Spirit of Ramanujan, Bruce C.Berndt, University Press
5. Introduction to Analytic Number Theory, Tom M.Apostol, University Press

IV. SUGGESTED CO-CURRICULAR ACTIVITIES:

1. Assignments
2. Student seminars
3. Quiz
4. Study Projects

VII Semester

Course 18: Network security & Cryptography

Credits -1

V. NETWORK SECURITY AND CRYPTOGRAPHY - PRACTICAL:

1. Write a python program that contains a string with a value ‘Hello world’. The program should XOR each character in this string with 0 and displays the result.
2. Write a Python program that contains a string with a value ‘Hello world’. The program should AND or and XOR each character in this string with 127 and display the result.
3. Write a python program to perform encryption and decryption using the Ceaser Cipher algorithm.

4. Write a python program to perform encryption and decryption using the Substitution algorithms:
5. Write a python program to perform encryption and decryption using the Hill Cipher algorithms:
6. Write a python a program for DES algorithm
7. Write a python a program for RSA algorithm
8. Write a python a program for Deffie Hellman algorithm
9. Write a python a program for SHA-I algorithm
10. Write a python a program for MD 5 algorithm

VIII Semester
Course 19: Data Science

Credits -3

LEARNING OUTCOMES:

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

1. Develop relevant programming abilities.

2. Demonstrate proficiency with statistical analysis of data.

3. Develop the ability to build and assess data-based models.

4. Demonstrate skill in data management

5. Apply data science concepts and methods to solve problems in real-world contexts and will communicate these solutions effectively

II. SYLLABUS:

UNIT I

Introduction: The Ascendance of Data, What is Data Science? , Finding key Connectors, Data Scientists You May Know, Salaries and Experience, Paid Accounts, Topics of Interest, Onward. Visualizing Data:matplotlib, Bar charts, Line charts, Scatterplots. Linear Algebra: Vectors, Matrices

UNIT II

Statistics: Describing a Single Set of Data, Correlation, Simpson’s Paradox, some Other Correlation Caveats, Correlation and Causation. Probability: Dependence and Independence, Conditional Probability, Bayes’s Theorem, Random Variables, Continuous Distributions, The Normal Distribution, The Central Limit Theorem. Hypothesis and Inference: Statistical Hypothesis Testing, Example: Flipping a Coin, Confidence Intervals, P-hacking, Example: Running an A/B Test, Bayesian Inference.

UNIT III

Getting Data: stdin and stdout, Reading Files – The Basics of Text Files, Delimited Files, Scraping the Web - HTML and the parsing Thereof, Example: O’Reilly Books About Data, Using APIs – JSON (and XML), Using an Unauthenticated API, Finding APIs. Working with Data: Exploring Your Data, Exploring One-Dimensional Data, Two Dimensions Many Dimensions, Cleaning and Munging, Manipulating Data, Rescaling, Dimensionality Reduction.

UNIT IV

Machine Learning: Modeling, What Is Machine Learning? Over fitting and under fitting, Correctness, The Bias-Variance Trade-off, Feature Extraction and Selection K-Nearest Neighbors: The Model, Example: Favorite Languages, The Curse of Dimensionality. Naive Bayes: A Really Dumb Spam Filter, A More Sophisticated Spam Filter, Implementation, Testing Our Model.

UNIT V

Decision Trees: What Is a Decision Tree? Entropy, The Entropy of a Partition, Creating a Decision

Tree, Putting It All Together, Random Forests. Neural Networks: Perceptron, Feed-Forward Neural Networks And Back propagation, Example: Defeating a CAPTCHA. Clustering: The Idea, The Model, Example: Meetups , Choosing k, Example: Clustering Colors, Bottom-up Hierarchical Clustering.

III. REFERENCES

1. Data Science from Scratch by Joel Grus O'Reilly Media

2. Wes McKinney, "Python for Data Analysis: Data Wrangling with Pandas, NumPy, and I Python", O'Reilly, 2nd Edition, 2018.

3. Jake VanderPlas, "Python Data Science Handbook: Essential Tools for Working with Data", O'Reilly, 2017.

4. Web resources:

   a. https://www.edx.org/course/analyzing-data-with-python

   b. http://math.ecnu.edu.cn/~lfzhou/seminar/[Joel_Grus]_Data_Science_from_Scratch_First_Princ.pdf

5. Other web sources suggested by the teacher concerned and the college librarian include reading material.

IV. SUGGESTED CURRICULAR ACTIVITIES:

1. Training of students by related industrial experts.
2. Assignments
3. Seminars, Group discussions, Quiz, Debates etc.(on related topics).

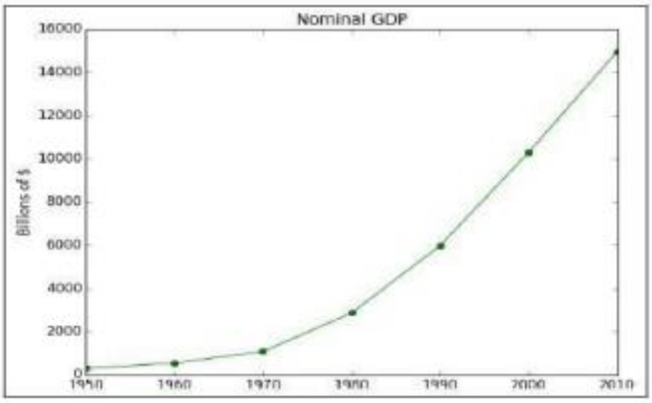
VIII Semester

Course 19: Data Science

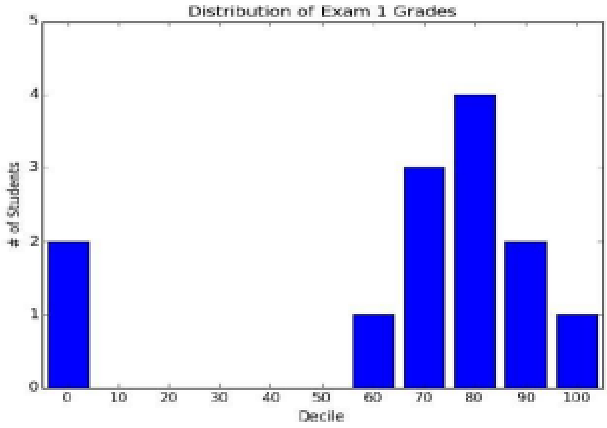
Credits -1

IV.DATA SCIENCE –PRACTICAL

1. Write a Python program to create a line chart for values of year and GDP as given below



2. Write a Python program to create a bar chart to display number of students secured different grading as given below



3. Write a Python program to create a time series chart by taking one year month wise stock data in a CSV file
4. Write a Python program to plot distribution curve
5. Import a CSV file and perform various Statistical and Comparison operations on rows / columns. Write a python program to plot a graph of people with pulse rate p vs. height h. The values of P and H are to be entered by the user.
6. Import rainfall data of some location with the help of packages available in R Studio and plot a chart of your choice.

VIII Semester

Course 20: Internet of Things

Credits -3

I. LEARNING OUTCOMES:

Students after successful completion of the course will be able to:
1. Appreciate the technology for IoT
2. Understand various concepts, terminologies and architecture of IoT systems.
3. Understand various applications of IoT
4. Learn how to use various sensors and actuators for design of IoT.
5. Learn how to connect various things to Internet.
6. Learn the skills to develop simple IOT Devices.

II. SYLLABUS:

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, BacnetAnd Modbus. IP Based Protocols for IoT: IPv6, 6LowPAN, LoRA, RPL, REST, AMPQ, CoAP, MQTT.

UNIT IV

Arduino Simulation Environment: Arduino Uno Architecture, Setting up the IDE, Writing Arduino Software, ArduinoLibraries, 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.

III. REFERENCES

1. Internet of Things - A Hands-on Approach, ArshdeepBahga and Vijay Madisetti, Universities Press, 2015, ISBN: 9788173719547

2. Vijay Madisetti and ArshdeepBahga, "Internet of Things (A Hands-onApproach)", 1st Edition, VPT, 2014

3. Daniel Minoli, ― "Building the Internet of Things with IPv6 and MIPv6: The

4. Evolving World of M2M Communications", ISBN: 978-1-118-47347-4, Willy i. Publications

5. Pethuru Raj and Anupama C. Raman, "The Internet of Things: Enabling Technologies, Platforms, and Use Cases", CRC Press

6. Open source software / learning websites

a. https://github.com/connectIOT/iottoolkit
b. https://www.arduino.cc/
c. https://onlinecourses.nptel.ac.in/noc17_cs22/course
d. http://www.cse.wustl.edu/~jain/cse570-15/ftp/iot_prot/index.html
e. Contiki (Open source IoT operating system)
f. Ardudroid (open source IoT project)
g. https://blynk.io(Mobile app)
h. IoT Toolkit (smart object API gateway service reference implementation)

7. Other web sources suggested by the teacher concerned and the college librarian including

reading material.

IV. SUGGESTED CURRICULAR ACTIVITIES:

1. Training of students by related industrial experts.
2. Assignments
3. Preparation and presentation of power-point slides, which include videos, animations,pictures, graphics, etc by the students.
4. Seminars, Group discussions, Quiz, Debates etc.(on related topics).
5. Field visits to identify the problems for IoT solutions

VIII Semester

Course 20: Internet of Things

Credits -1

V.INTERNET OF THINGS –PRACTICAL

1. Installing and work with Arduino IDE
2. Blinking LED sketch with Arduino
3. Simulation of 4-Way Traffic Light with Arduino
4. Using Pulse Width Modulation
5. LED Fade Sketch and Button Sketch
6. Analog Input Sketch (Bar Graph with LEDs and Potentiometer)
7. Digital Read Serial Sketch (Working with DHT/IR/Gas or Any other Sensor)
8. Working with Adafruit Libraries in Arduino
9. Spinning a DC Motor and Motor Speed Control Sketch

10. Working with Shields
11. Design APP using Blink App or Things peak API and connect it LED bulb.
12. Design APP Using Blynk App and Connect to Temperature, magnetic Sensors.

VIII Semester
Course 21: Mean Stack Development

Credits -3

I.LEARNING OUTCOMES:

Students after successful completion of the course willable to:
1. Understand the advanced Java Script Concepts
2. Develop Node.js: a server-side JavaScript run-time Application
3. Implement Express.js: a server-side JavaScript framework running on top of Node.js.
4. Implement Angular: a browser-independent MVC JavaScript UI framework.
5. Develop MongoDB-a schema-less (document-oriented) NoSQL database. 6.Apply Deployment Techniques & Working with cloud platform

II.SYLLABUS

UNITI

Basic Web DevelopmentFramework,Node.js – to-Angular Stack Components JavaScript Primer: Defining Variables, Understanding JavaScript Data Types, Operators, Looping, Creating Functions, Variable Scope, JavaScript Objects, Manipulating Strings, Working with Arrays, Adding Error Handling, Events and Document Object Model, Handling JSON 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

UNITIII

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.

UNITIV

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.

UNITV

Building Single Page Applications with AngularJS Single Page Application – Introduction, Two-way data binding(Dependency Injection), MVC in Angular JS, Controllers, Getting user input, 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 digest loop and use of $apply.

III.REFERENCES:

TEXT BOOK(S)
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”, Peach pit press, 2015
2. Ken Williamson, “Learning Angular JS”, O′Reilly; 1 edition
3. Mithun Satheesh, “Web development with MongoDB and Node JS”, Pack Publishing Limited; 2nd Revised edition.
4. Web Links: https://www.geeksforgeeks.org/introduction-to-mean-stack/
IV.SUGGESTEDCO-CURRICULARACTIVITIES:
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 WebSocket.
5. Build real timed ash board in MEAN stack using WebSocket
6. Develop a CURD APP for College Student Database

VIII Semester

Course 21: Mean Stack Development

Credits -1

V.MEAN STACK DEVELOPMENT- PRACTICAL

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.