GODVARI INSTITUTE OF ENGINEERING & TECHNOLOGY(A)

Department of Computer Science & Engineering

COURSE STRUCTURE

B. Tech. Computer Science & Engineering

II Year **I Semester**

S. No.	Subject Title	Periods per week			C	Scheme of Examination Maximum Marks		
		T	P	D		Int.	Ext.	Total
14159301	Managerial Economics and Financial Analysis	4	-	-	3	30	70	100
14150302	Object Oriented Programming through C++	4	-	-	3	30	70	100
14150303	Mathematical Foundations of Computer Science	4	-	-	3	30	70	100
14154304	Digital Logic Design	4	-	-	3	30	70	100
14150305	Data Structures	4	-	-	3	30	70	100
14150311	Object Oriented Programming Lab	-	3	-	2	50	50	100
14150312	Data Structures Lab	-	3	-	2	50	50	100
14154313	Digital Logic Design Lab	-	3	-	2	50	50	100
14159392	Soft Skills – 1#	3	1	-	-	-	-	-
Total		**23**	**10**	**0**	**21**	**300**	**500**	**800**

II Year **II Semester**

S. No.	Subject Title	Periods per week			C	Scheme of Examination Maximum Marks		
		T	P	D		Int.	Ext.	Total
14159401	Probability and Statistics	4	-	-	3	30	70	100
14150402	Java Programming	4	-	-	3	30	70	100
14150403	Advanced Data Structures	4	-	-	3	30	70	100
14150404	Formal Languages and Automata Theory	4	-	-	3	30	70	100
14150405	Computer Organization	4	-	-	3	30	70	100
14150411	Advanced Data Structures Lab	-	3	-	2	50	50	100
14150412	Java Programming Lab	-	3	-	2	50	50	100
14150413	Free Open Source Software (FOSS) Lab	-	3	-	2	50	50	100
Total		**20**	**9**	**0**	**21**	**300**	**500**	**800**

II Year B.Tech. (CSE). – I Semester

Managerial Economics and Financial Analysis

Unit-I

Objective: To understand the concept; and nature of Managerial Economics and its relationship with other disciplines, concepts of Demand and Demand forecasting for Proper Production Planning.

Introduction to Managerial Economics and demand Analysis: Definition of Managerial Economics and Scope – Managerial Economics and its relation with other subjects – Concepts of Demand – Types – Determinants, Law of Demand its Exception – Elasticity of Demand – Types and Measurement - Demand forecasting and its methods.

Unit-II

Objective: To understand the concept of Production function, Input Output relationship, different Cost Concepts and Concept of Cost – Volume – Profit Analysis.

Production and Cost Analysis: Production function Isoquants and Isocosts – Law of Variable proportions – Cobb-Douglas Production function- Economies of Scale- Cost Concepts- Opportunity Cost-Fixed Vs Variable Costs – Explicit Cost Vs Implicit Costs – Out of Pocket Costs Vs Imputed Costs – Cost Volume Profit Analysis- Determination of Break-Even Point (Simple Problems)

Unit-III

Objective: To understand the nature of Competition, Characteristics of Pricing in the different market structure and significance of various pricing methods.

Introduction to Markets, Theories of the Firm and Pricing Policies: Market Structures: Perfect Competition, Monopoly and Monopolistic and Oligopoly – Features – Price, Output Determination – Managerial Theories of firm: Maris and Williamson's models – Methods of Pricing: Limit Pricing, Market Skimming Pricing, And Internet Pricing: Flat Rate Pricing, Usage sensitive, Transaction based pricing, Priority Pricing.

Unit-IV

Objective: To know the different forms of Business Organization and their Merits and Demerits both Public and Private Enterprises and the concepts of Business Cycles.

Types of Business Organizations and Business Cycles: Features and Evaluation of Sole trader – Partnership – Joint Stock Company – State / Public Enterprises and their forms – Business Cycles – Meaning and Features – Phases of Business Cycle.

Unit-V

Objective: To understand the different Accounting Systems preparation of Financial Statements and uses of Different tools for performance evaluation.

Introduction to Financial Accounts: Introduction to Double Entry Systems, Preparation of Journal – Subsidiary Books- Ledger-Cash Book-Trial Balance- Preparation of Financial Statements, Analysis of Financial Statements through Ratio Analysis (Simple Problems).

Unit -VI

Objective: To understand the concept of Capital, Capital Budgeting and to know the techniques used to evaluate Capital Budgeting proposals by using different methods.

Capital, Capital Budgeting: Capital, Significance of Capital, Sources of Finance (Capital) - Meaning of Capital Budgeting Need for Capital Budgeting - Techniques of Capital Budgeting - Traditional and Modern Methods.

Text Books:

1. Prof. J.V. Prabhakara Rao, Prof.P. Venkata Rao. "Managerial Economics and Financial Analysis", Ravindra Publication.
2. Dr.A.R.Aryasri- Managerial Economics and Financial Analysis – TMH Publications.
3. Dr.N.Appa Rao, Dr.P. Vijay Kumar 'Managerial Economics and Financial Analysis", Cengage Publications New Delhi

Reference Books:

1. Dr.B. Kuberudu and Dr.T.V. Ramana: Managerial Economics & Financial Analysis, Himalaya Publishing House.

II Year B.Tech. (CSE). – I Semester

Object-Oriented Programming through C++

Unit-I

Objectives: Exposure to basics of object oriented mode, C++ programming and I/O in C++

Introduction: Differences between C and C++, the Object Oriented Technology, Disadvantage of Conventional Programming, Concepts of Object Oriented Programming, Advantages of OOP, Structure of a C++ Program, Header Files and Libraries.

Input and Output in C++: Introduction, Streams In C++ and Stream Classes, Pre-Defined Streams, Stream Classes, 'Formatted and Unformatted Data, Unformatted Console I/O Operations, Member Functions of Istream Class, Formatted Console I/O Operations.

Unit-II

Objectives: Focus on Basic concept in C++ programming, Operators, control structures, functions, overloading, recursion

Tokens In C++, Variable Declaration and Initialization, Data Types, Operators In C and C++, Scope Access Operator, Namespace, Memory Management Operators, Comma Operator, Revision of Decision Statements, Control Loop Statements

Functions in C++: Introduction, Structure of Function, Passing Arguments, retrun by Reference, Returning More Values by Reference, Inline Functions

Unit-III

Objectives: Acquaintance with classes, objects and member functions

Classes and Objects: Introduction, Classes in C++, Declaring Objects, Access Specifiers and Their Scope, Member Functions, Outside Member Function as Inline, Data Hiding or Encapsulation, Classes, Objects and Memory, Static Member Variables, Static Member Functions Static Object, Array of Objects, Friend Functions, the Const Member Functions, the Volatile Member Function, Recursive Member Function. Member Function and Non- Member Function, Overloading Member Functions, Nested Classes.

Unit-IV

Objectives: Focus on constructors, destructors, variants in them, operator overloading, type Conversions

Constructors and Destructors: Introduction, Characteristic of Constructors & Destructors, Applications with Constructors, Parameterized Constructor, Overloading Constructors (Multiple Constructors), Array of Objects Using Constructors, Constructors with Default Arguments, Copy Constructors, the Const Objects, Destructors, Calling Constructors and Destructors.

Operator Overloading: Introduction, Overloading Unary Operators, Constraint on Increment and Decrement Operators, Overloading Binary Operators, Overloading with Friend Function, Overloading Assignment Operator (=).

Unit-V

Objective: Concentration on inheritance, types of inheritance, polymorphism, virtual functions

Inheritance: Introduction, Reusability, Access Specifies and Simple Inheritance, Protected Data with Private Inheritance, Types of Inheritances (Single Inheritance, Multilevel Inheritance, Multiple Inheritance, Hierarchical Inheritance, Hybrid Inheritance, Multipath Inheritance), Virtual Base Classes, Constructors, Destructors, and Inheritance, Object as a Class Member, Abstract Classes, Constructor in Derived Class, Pointers and Inheritance, Overloading Member Function, Advantages of Inheritance, Disadvantages of Inheritance.

Unit-VI

Binding, Polymorphisiv and Virtual Functions: Introduction, Binding In CH, Static (Early) Binding, Dynamic (Late) Binding, Pointer to Base and Derived Class Objects, Virtual Functions, Rules for Virtual Functions, Array of Pointers, Pure Virtual Functions, Abstract Classes, Working of Virtual Functions, Virtual Functions In Derived Classes, Object Slicing, Constructors and Virtual Functions, Virtual Destructors, Destructor and Virtual Functions.
Exception Handling: Introduction, Principles of Exception Handling, The key words Try, Throw and Catch, Exception Handling Mechanism, Multiple Catch Statements, Catching Multiple Exceptions, Re-Throwing Exception.

Text Books:

1. Programming in C++, Ashok N Kamthane. Pearson 2nd Edition.
2. The Complete Reference, C++, 4ed, Herbert Schildt, TMH

Reference Books:

1. Object oriented programming with C++, 3rd ed, E. Balaguruswamy
2. Object Oriented Programming with C++, 2nd ed, Sourav Sahay, OXFORD.
3. Object Oriented Programming C++, Joyce Farrell, Cengage
4. Mastering C ++, Venugopal, Rajkumar, Ravi kumar TMH.

II Year B.Tech. (CSE). – I Semester

Mathematical Foundations of Computer Science

Unit-I

Objective: Acquiring the relevance of statements, inferences and predicates in computer science

Mathematical Logic

Propositional Calculus: Statements and notations, Connectives, Well formed formulas, Truth Tables, tautologies, equivalence of Formulas, Tautological Implications, Normal forms, Argument.

Predicate Calculus: Predicative Logic, statement functions, variables and quantifiers, free & bound variables, inference theory of predicate calculus

Unit-II

Objective: Focuses on sets and their relations and their operations, relations on functions

Set Theory: Introduction, binary sets, operations on binary sets, Principle of Inclusion and Exclusion.

Relations: Introduction, Properties of Binary Relations, Operations on Relations, Equivalence, Transitive closure, Compatibility Relations, Partial ordering Relations, Lattices, Hasse

diagram.

Functions: Introduction, Types of functions, Composition of functions, Inverse functions, Recursive functions, Pigeonhole principle.

Unit-III

Objective: Exposure of Graphs their representation types, trees and tree variants.

Graph Theory: Basic Concepts, Matrix Representation of Graphs: Adjacency Matrices, Incidence Matrices, Types of Graphs.

Spanning Trees: Properties, Algorithms for Spanning trees and Minimum Spanning Tree

Unit-IV

Objective: Overview of number theory, basic algorithms in number theory.

Number Theory & Induction: Properties of integers, Division Theorem, The Greatest Common Divisor, Euclidean Algorithm, The Fundamental Theorem of Arithmetic, Modular Arithmetic (Fermat's Theorem and Euler's Theorem)

Unit-V

Objective: Overview of algebraic structures, group theory, binomial theorem

Algebraic Structures: Lattice: Properties, Lattices as Algebraic Systems, Algebraic Systems with one Binary Operation, Properties of Binary operations Groups: Abelian Group, Cosets, Subgroups (Definitions and Examples of all Structures)

Binomial Theorem: Binomial and Multinomial Coefficients, Generating functions, Properties of Generating functions, The Principles of Inclusion – Exclusion.

Unit-VI

Objective: Overview of recurrence relation and solving recurrence relation

Recurrence Relation: Recurrence Relations, Formulation as Recurrence Relations, Solving linear homogeneous recurrence Relations by substitution, generating functions and the Method of Characteristic Roots, Solving Inhomogeneous Recurrence Relations.

Text Books:

1. Discrete Mathematics, Swapan Kumar chakrborthy, Bikash kanti sarkar, OXFORD.
2. Discrete Mathematical Structures with Applications to Computer Science, Tremblay, Manohar, TMH.

Reference Books:

1. Discrete Mathematics, Proofs, Structures and applications, 3rd ed, CRC Press
2. Discrete Mathematics, S.Santha, Cengage
3. Discrete Mathematics with Applications, Thomas Koshy, Elsevier

II Year B.Tech. (CSE). – I Semester

Digital Logic Design

Course Objective:

- ***Introduce the concept of digital and binary systems***
- ***Be able to design and analyze combinational logic circuits.***
- ***Be able to design and analyze sequential logic circuits.***
- ***Understand the basic software tools for the design and implementation of digital circuits and systems.***
- ***Reinforce theory and techniques taught in the classroom through experiments and projects in the laboratory***

Unit-I

Number Systems: Binary, Octal, Hex Decimal, and Conversions, range; Binary additions and subtractions (using 1c, and 2c), concept of overflow; representations of negative numbers using 1's and 2's complement and range; BCD numbers**:** Representation of 8421, 2421, Ex-3, Gray and self complementary codes; additions and subtractions on 8421 codes; Error detecting codes**:** even, odd parity, hamming codes; Error correcting codes**:** hamming codes, block parity codes; Floating point representation.

Unit-II

Logic Gates and Boolean Algebra: Boolean Algebra and Digital Logic GATES, Basic Boolean LAWs and properties; Boolean functions; canonical and standard forms (SOP, POS); Gate minimization using three and four variable K-Map's with and without don't cares. Encoders, Decoders, Multiplexers, D-Multiplexers;

Unit-III

Combinational Logic Circuits: Definition of combinational circuits, design procedure for half, full, decimal (8421) adders and subtractors; Combinational Circuit Design for BCD code converters;

Unit-IV

Sequential Logic Circuits: Classification of Sequential circuits, latches, Flip Flops; Analysis of clocked sequential circuits, State Reduction and Assignment, Register, Ripple Counters, Synchronous Counters, Other Counters.

Unit-V

Introduction to Memories: Types of Memories – Main memory – random access memory, ROM, Types of ROM; Decoder and RAM interface: Address lines, data lines, chip select signal; Cache Memory- design issues, hit and miss ratio related problems; secondary memory types ; Associative and Auxiliary memory; Memory hierarchy in terms of speed and capacity.

Unit-VI

Programmable Logic Devices: PLA, PAL, PROM. Realization of Switching Functions Using

PROM, PAL and PLA. Comparison of PLA, PAL and PROM.

Text Books:

1. Digital Design – Third Edition, M. Morris Mano, Pearson Education/PHI.
2. Fundamentals of Logic Design, Roth, Fifth Edition, Thomson.

Reference Books:

1. Switching and Finite Automata Theory,3/e,Kohavi, Jha, Cambridge.
2. Digital Logic Design, Leach, Malvino, Saha,TMH
3. Modern Digital Electronics, R.P. Jain, TMH .

II Year B.Tech. (CSE). – I Semester

Data Structures

Unit-I

Objective: Exposure to Recursive algorithms

Introduction: Definition, types of data structures: Linear and non linear data structures, Static and Dynamic representation of data structures and comparison.

Recursion: Definition, Design methodology and Implementation of recursive algorithm, Linear and Binary recursion, Recursive algorithm for Factorial function, GCD computation, Fibonacci sequence, Towers of Hanoi, Tail recursion.

Unit-II

Objective: Applying stack and queue techniques for logical operation

Stacks: Definition, Basic stack operations, Representation of a stack using arrays, stack applications: reverse list, factorial calculations, infix to postfix transformations, evaluating arithmetic expressions.

Queues: Definition, Basic queue operations, representation of queue using arrays, implementation of queue using stack, applications of queue: Dequeue, enqueue, circular queue.

Unit-III

Objective: Exposure to list representation models in various types of applications

Linked Lists: Introduction of linked list, types of linked lists, representation of single linked list in memory, operations on single linked list, merge two single linked list into one list, representation of stacks and queue using linked list, represent polynomial expression using linked list, advantages and disadvantages of single linked list, circular single linked list, Double linked list.

Unit-IV

Objective: Implementation of tree implementation various forms advanced understating of other variants of trees and their operations.

Trees: Basic tree concepts, Binary tree: Properties, representation of binary trees using arrays and linked lists, binary tree operations, Binary tree traversals: in-order, pre-order, post-order
Binary Search Trees: Properties, BS Trees operations: insertion, deletion, threaded binary trees.
Advanced concepts of Trees: Introduction of balanced trees (AVL trees).

Unit-V

Objective: orientation on Graphs, representation of graphs, Graph traversals, spanning trees

Graphs: Introduction of graphs, representation of graphs using linked list and adjacency matrix, graphs operations and algorithms: insert an edge, delete an edge, insert a node, and delete a node. Graph traversal algorithms: Breadth First Search algorithms and Depth First Search algorithm.
Spanning Trees: Minimum cost spanning tree, Kruskal's Algorithm, Prim's algorithm.

Unit-VI

Objective: Exposure to algorithmic complexities, sorting techniques

Data Abstraction Analysis: Time complexity and space complexity, Asymptotic Notation Big O, Omega, and Theta notations.
Sortings: Quick sort, heap sort, merge sort.

Text Books:

1. Data structures using C, 2nd Edition, Reema Thareja, Oxford higher education
2. Data structures: A Pseudocode Approach with C, 2nd edition, R.F.Gilberg and B.A.Forouzan, Cengage Learning

Reference Books:

1 Data Structures with C, Seymour Lipscutz, Schaum's Outlines, TMH-special 2nd Edition
2. Data structures A Programming Approach with C, D.S.Kushwaha and A.K.Misra, PHI.
3. Data structures and Algorithm Analysis in C, 2nd edition, M.A.Weiss, Pearson.
4. Data Structures using C, A.M.Tanenbaum,Y. Langsam, M.J.Augenstein, Pearson.
5. Data structures and Program Design in C, 2nd edition, R.Kruse, C.L.Tondo and B.Leung, Pearson.

II Year B.Tech. (CSE) – I Semester

Object-Oriented Programming Lab

1. Write a C++ program illustrating Variable Scope.
2. Write a C++ program illustrating Swap integer values by reference.
3. Write a C++ program illustrating Checking whether the number is even or odd using Ternary operator.
4. Write a C++ program illustrating a program to find the roots of a quadratic equation .Use switch statements to handle different values of the discriminant (b^2-4*a*c).
5. Write a C++ program illustrating interactive program to multiply 2 variables after checking the compatibility.
6. Write a C++ program illustrating interactive program for computing the roots of a quadratic equation by handling all possible cases. Use streams to perform I/O operations.
7. Write a C++ program illustrating to sort integer numbers.
8. Write a C++ program illustrating factorial using recursion.
9. Write a C++ program illustrating pass by value, pass by reference, pass by address.
10. Write a C++ program illustrating Function overloading.
11. Write a C++ program illustrating an interactive program for swapping integer, real, and character type variables without using function overloading .Write the same program by using function overloading features and compare the same with its C counterpart.
12. Write a C++ program illustrating inline functions.
13. Write a C++ program illustrating Friend function.
14. Write a C++ program illustrating Exception handling.
15. Write a C++ program illustrating Function template.
16. Write a C++ program illustrating Overloading increment, decrement, binary+&<< operator.
17. Write a C++ program illustrating Virtual function.
18. Write a C++ program illustrating an interactive program to process complex numbers. It has to perform addition, subtraction, multiplication, and division of complex numbers. Print results in x+iy form. Create a class for the complex number representation.
19. Write a C++ program illustrating user defined string processing functions using pointers (string length, string copy, string concatenation)
20. Write a C++ program illustrating Constructor overloading (Both parameterised and default).
21. Write a C++ program illustrating Copy constructor.
22. Write a C++ program illustrating access data members & member functions using ‘THIS’ pointer.
23. Write a C++ program illustrating for overloading ++ operator to increment data.
24. Write a C++ program illustrating overloading of new and delete operator.
25. Write a C++ program illustrating Abstract classes.
26. Write a C++ program illustrating Inheritance (Multiple, Multilevel, Hybrid).
27. Write a C++ program illustrating Virtual classes & virtual functions.
28. Write a C++ program illustrating overloading function template.

II Year B.Tech. (CSE)– I Semester

Data Structures Lab

Exercise-1

1. Write recursive program for the following
2. Write recursive and non recursive C program for calculation of GCD (n, m)
3. Write recursive and non recursive C program for Towers of Hanoi: N disks are to be transferred from peg S to peg D with Peg I as the intermediate peg.

Exercise-2

1. Write a C program that use recursive function to perform Binary Search for a key value in a given list.
2. Write a C program that use recursive function to perform Linear Search for a key value in a given list.

Exercise-3

1. Write C program that implement Quick sort, to sort a given list of integers in ascending order
2. Write C program that implement heap sort, to sort a given list of integers in ascending order
3. Write C program that implement merge sort, to sort a given list of integers in ascending order

Exercise-4

1. Write C program that implement stack (its operations) using arrays
2. Write C program that implement stack (its operations) using Linked list

Exercise-5

1. Write a C program that uses Stack operations to Convert infix expression into postfix expression
2. Write C program that implement Queue (its operations) using arrays.
3. Write C program that implement Queue (its operations) using linked lists

Exercise-6

1. Write a C program that uses functions to create a singly linked list
2. Write a C program that uses functions to perform insertion operation on a singly linked list
3. Write a C program that uses functions to perform deletion operation on a singly linked list

Exercise-7

1. Write a C program to reverse elements of a single linked list.
2. Write a C program to store a polynomial expression in memory using linked list

Exercise-8

1. To Implement operations on graph
    a. Vertex insertion, Vertex deletion, Finding vertex, Edge addition and deletion
2. To implement depth fist search for a graph non recursively.
3. To implement breadth fist search for a graph non recursively.

Exercise-9

1. Write a C program to Create a Binary Tree of integers
2. Write a recursive C program for Traversing a binary tree in preorder, inorder and postorder.

Exercise-10

1. Write a C program to Create a BST
2. Write a C program to delete a node from a BST.
3. Write a C program to insert a node into a BST.

Exercise-11

1. To implement prim's Algorithm to generate a min-cost spanning tree.
2. To implement krushkal's Algorithm to generate a min-cost spanning tree.

II Year B.Tech. (CSE)– I Semester

Digital Logic Design Lab

1. Verification of Basic Logic Gates.
2. Implementing all individual gates with Universal Gates NAND & NOR.
3. Design a circuit for the given Canonical form, draw the circuit diagram and verify the De-Morgan laws.
4. Design a Combinational Logic circuit for 4x1 MUX and verify the truth table.
5. Design a Combinational Logic circuit for 1x4 De- MUX and verify the truth table.
6. Implement Binary to Gray Code Conversion and Vice Versa.
7. Construct 2-Bit Comparator using Gates.
8. Construct Half subtractor and Full Subtractor using Half Subtractor & Verify the truth Tables.
9. Construct Half Adder and Full Adder using Half Adder and verify the truth table.
10. Verification of truth tables of the basic Flip- Flops with Synchronous and Asynchronous modes.
11. Construct 2x4 Encoder and 4 x 2 Decoder using Logic Gates
12. Design a Decade Counter and verify the truth table.
13. Design the Mod 6 counter using D-Flip -Flop.
14. Design a 4 – bit right Shift Register using D-Flip -Flop and verify the truth table.

II Year B.Tech. (CSE). – I Semester

Soft Skills-1

(Title: Professional Communication-1)

Course Objective: To strengthen the four language skills of the learners and to prepare them for success in academics and the job market.

Semester I **60 hours + Assessment 1 hrs**

Total: 15 weeks Theory: 45 hours Practical: 15 hours

Per week: 4 hours Theory: 3 hours Practical: 1 hours

Unit-1: Places		**Theory/Lab**	**Time frame**
Reading	Introducing the theme; Scanning for information/numbers; understanding key vocab; making predictions	Punctuation Understanding unknown text	
Writing	Punctuation, sentence structure, writing different sentences		
Listening & Pronunciation	listening for main ideas/details; wordstress vowel sounds	Lab	
Speaking	Organizing information for a presentation*;* Making a presentation	Lab	
Grammar	Parts of speech: Nouns, verbs and adjectives; Subject + verb; There is / There are; Past simple		
Vocabulary	Vocabulary to describe places		

Unit-2: Festivals And Celebrations		**Theory/Lab**	**Time frame**
Reading	Previewing a text using the title, sub-titles and photographs; recognizing text types, skimming the text	Pre-reading requires viewing of a video	
Writing	Organizing sentences into a paragraph*;* writing a first draft; writing paragraph :descriptive ,narrative etc.		
Listening & Pronunciation	Listening and taking notes; listening for examples; Stressed words and unstressed sounds	Lab	
Speaking	Making suggestions*;* Giving a poster presentation, understanding intonation	Lab	
Grammar	Prepositions of time and place: *on, in, at;* Adverbs of frequency*;* Sentence structure: subject and verb order; Prepositional phrases; Present tense question forms		
Vocabulary	Vocabulary to describe festivals; Collocations		

Unit-3: School And Education		Theory/Lab	Time frame
Reading	Skimming for main ideas; reading for details; making inferences	Pre-reading requires viewing of a video	
Writing	Paragraph organization: topic sentence and supporting sentences, selection of type of paragraph, ordering and sequencing		
Listening & Pronunciation	Using visual clues to listen; following native accent and intonation	Lab	
Speaking	Giving opinions in a debate*:* agreeing and disagreeing, convincing	Lab	
Grammar	Tense and aspect, use of *because* and *so;* basic verb patterns		
Vocabulary	Vocabulary in academic context; Collocations about learning; Prepositional phrases		

Unit-4: The Internet And Technology		Theory/Lab	Time frame
Reading	Understanding theme; Scanning to predict content; Making inferences	Pre-reading requires viewing of a video	
Writing	Describing an ordering steps, structurizing information		
Listening & Pronunciation	Listening for reasons; Listening and predicting the inner theme and conclusion ,consonant sounds	Lab	
Speaking	Presenting additional or contrasting information;	Lab	
Grammar	Compound nouns; *and, also* and *too*; *but* and *however; can* / *be able to*		
Vocabulary	Vocabulary for Internet and technology		

Unit-5: Language And Communication		Theory/Lab	Time frame
Reading	Reading for main ideas, identifying the meaning, preparing captions	Pre-reading requires viewing of a video	
Writing	Writing supporting sentences; Reviewing a paragraph for content and structure, report writing ,types of report		
Listening & Pronunciation	Listening for genre; Listening for instructions; Consonant sounds	Lab	
Speaking	Sequencing words to organize instructions; Planning and giving a set of instructions	Lab	
Grammar	Countable and uncountable nouns; Articles *a, an* or no article; Quantifiers: *some, many, a lot of, a few, a little;* Imperative clauses; Verb patterns		
Vocabulary	Vocabulary for every day communication		

II Year B.Tech. (CSE). – II Semester

Probability & Statistics

Unit-I

Probability

Objective: To impart the basic concepts of Probability with the help of theorems

Introduction: Sample point, sample space, event, mutually exclusive, independent and exhaustive events, probability axioms, addition theorem, multiplication theorem, and other basic theorems on probability, conditional probability, pair-wise independence, Baye's theorem

Unit-II

Random variables and Distributions

Objective: To gain the knowledge of random variables and its types with respective distributions as modern tools for engineering practices

Introduction- Random variables- Distribution function- Discrete distributions - Binomial and Poisson distributions - Continuous distributions: Normal, Normal approximation to Binomial distribution, Gamma and Weibull distributions

Unit-III

Moments and Generating functions

Objective: To understand and identify the generating functions of various distributions and solve engineering problems

Introduction-Mathematical expectation and properties - Moment generating function - Moments of standard distributions (Binomial, Poisson and Normal distributions) – Properties.

Unit-IV

Sampling Theory

Objective: To be familiar with types of sampling and estimation techniques and impart problem – solving skills in various engineering Applications

Introduction - Population and samples- Sampling distribution of mean for large and small samples (with known and unknown variance) – Proportion sums and differences of means - Sampling distribution of variance –Point and interval estimators for mean and proportions- Chebyshev's Inequality.

Unit-V

Tests of Hypothesis

Objective: To understand the multi-disciplinary inferential statistics, design of experiments and try to find out the solutions for global economical, social and economical issues

Introduction - Type I and Type II errors - Maximum error - One tail, two-tail test - Tests concerning one mean and proportion, two means- Proportions and their differences using Z-test, Student's t-test - F-test and Chi –square test - ANOVA for one-way and two-way classified data.

Unit-VI

Curve fitting and Correlation

Objective: To understand and analyze fitting of linear and non-linear with statistical tools and impart techniques for Practicing the correlation – regression by identifying the formulae

Introduction - Simple Correlation and Regression - Rank correlation - Multiple regression-Fitting a straight line -Second degree curve exponential curve- power curve by method of least squares.

Text Books:

1. Probability and Statistics for Engineers: Miller and John E. Freund, Prentice Hall of India
2. Probability and Statistics for Engineers and Scientists: Ronald E. Walpole, Sharon L. Mayers and Keying Ye: Pearson
3. Probability, Statistics and Random Processes, Murugesan, Anuradha Publishers, Chennai.

II Year B.Tech. (CSE). – II Semester

JAVA Programming

Unit-I

Introduction to OOP: Introduction, Need of Object Oriented Programming, Principles of Object Oriented Languages, Procedural languages vs OOP, Applications of OOP

Java Basics: History of JAVA, Java Virtual Machine, Java Features, Program structures.

Unit-II

Programming Constructs: Variables, Primitive Datatypes, Identifiers- Naming Conventions, Keywords, Literals, Operators-Binary, Unary and ternary, Expressions, Precedence rules and Associativity, Primitive Type Conversion and Casting, Flow of control-Branching, Conditional, loops.

Classes and Objects: classes, Objects, Creating Objects, Methods, constructors-Constructor overloading, Cleaning up unused objects-Garbage collector, Class variable and Methods- Static keyword, this keyword, Arrays, Command line arguments.

Unit-III

Inheritance: Types of Inheritance, Deriving classes using extends keyword, Method overloading, super keyword, final keyword, Abstract class.

Interfaces & Packages: Interface-Extending interface, Interface Vs Abstract classes, Packages-Creating packages, using Packages, Access protection, java.lang package.

Exceptions & Assertions - Introduction, Exception handling techniques-try...catch, throw, throws, finally block, user defined exception, Exception Encapsulation and Enrichment, assertions

Unit -IV

Multithreading: java.lang.Thread, The main Thread, Creation of new threads, Thread priority, Multithreading- Using isAlive() and join(), Syncronization, suspending and Resuming threads, Communication between Threads.

Input/Output: reading and writing data, java.io package

Unit-V

Applets: Applet class, Applet structure, An Example Applet Program, Applet Life Cycle, paint(), update() and repaint()

Event Handling: Introduction, Event Delegation Model, java.awt.event, Description, Sources of Events, Event Listeners, Adapter classes, Inner classes

Unit-VI

Abstract Window Toolkit: Why AWT?, java.awt package, Components and Containers, Button, Label, Checkbox, Radio buttons, Listboxes, Choice boxes, Text field and Text area, container classes, Layouts, Menu, Scroll bar.
Swing: Introduction, JFrame, JApplet, JPanel, Components in swings, Layout Managers, JList and JScroll Pane, Split Pane, JTabbedPane, Dialog Box Pluggable Look and Feel
JDBC Connectivity: JDBC type 1 to 4 drivers, connection establishment, Query Execution

Text Books:

1. The Complete Reference Java, 8ed, Herbert Schildt, TMH
2. Programming in JAVA, Sachin Malhotra, Saurabh choudhary, Oxford.

Reference Books:

1. Programming in Java E.BalaguruSwamy.
2. JAVA Programming, K.Rajkumar.Pearson
3. Core JAVA, Black Book, Nageswara Rao, Wiley, Dream Tech
4. Core JAVA for Beginners, Rashmi Kanta Das, Vikas.
5. Object Oriented Programming Through Java, P. Radha Krishna, Universities Press.

II Year B.Tech. (CSE). – II Semester

Advanced Data Structures

(Note: C++ and Java implementation is not included in the syllabus)

Unit-I

Objectives: Comprehensive understanding of dictionaries, hashing mechanism which supports faster retrieval and skip lists

Dictionaries: Sets, Dictionaries, Hash Tables, Open Hashing, Closed Hashing (Rehashing Methods), Hashing Functions (Division Method, Multiplication Method, Universal Hashing), Skip Lists, Analysis of Skip Lists. (Reference-1)

Unit-II

Objectives: llustration of balanced trees and their operations

AVL Trees: Maximum Height of AVL Tree, Insertions and Deletions. 2-3 Trees: Insertion, Deletion.

Unit-III

Objectives: Comprehension of heaps, queues and their operations Priority Queues:

Binary Heaps: Implementation of Insert and Delete min, Creating Heap.

Binomial Queues: Binomial Queue Operations, Binomial Amortized Analysis, Lazy Binomial Queues

Unit-IV

Objectives: Detailed knowledge of nonlinear data structures and various algorithms using them Graph algorithms: Minimum-Cost Spanning Trees- Prim's Algorithm, Kruskal's Algorithm, Shortest Path Algorithms: Dijkstra's Algorithm, All Pairs Shortest Paths Problem: Floyd's Algorithm, Warshall's Algorithm,

Unit-V

Objectives: Analysis of complexities in various sorting techniques along with their lower bounds Sorting Methods: Order Statistics: Lower Bound on Complexity for Sorting Methods: Lower Bound on Worst Case Complexity, Lower Bound on Average Case Complexity, Heap Sort, Quick Sort, Radix Sorting, Merge Sort.

Unit-VI

Objectives: Illustration of tries which share some properties of table look up, various issues related to the design of file structures

Pattern matching and Tries: Pattern matching algorithms- the Boyer –Moore algorithm, the Knuth-Morris-Pratt algorithm

Tries: Definitions and concepts of digital search tree, Binary trie, Patricia, Multi-way trie

Text Books:

1. Data Structures, A Pseudocode Approach, Richard F Gilberg, Behrouz A Forouzan
2. Data structures and Algorithm Analysis in C, 2nd edition, Mark Allen Weiss, Pearson
3. Fundamentals of Data Structures in C: 2nd ed, , Horowitz , Sahani, Anderson-freed, Universities Press

Reference Books:

1. Web : http://lcm.csa.iisc.ernet.in/dsa/dsa.html
2. http://utubersity.com/?page_id=878
3. http://freevideolectures.com/Course/2519/C-Programming-and-Data-Structures
4. http://freevideolectures.com/Course/2279/Data-Structures-And-Algorithms
5. File Structures :An Object oriented approach with C++, 3rd ed, Michel J Folk, Greg Riccardi, Bill Zoellick
6. C and Data Structures: A Snap Shot oriented Treatise with Live examples from Science and Engineering, NB Venkateswarlu & EV Prasad, S Chand, 2010.

II Year B.Tech. (CSE). – II Semester

Formal Languages and Automata Theory

Course objectives:

1. ***Learn the fundamental concepts of formal languages and automata.***
2. ***Be able to construct regular expressions, grammars, and automata for different levels of formal languages.***
3. ***Understand the relationship between Turing machines and modern computers.***
4. ***Understand the limitations and undecidable problems of modern computers.***

Unit-I

Fundamentals: Symbols, Alphabets, Strings, Languages, Operations on strings and languages, Finite state machine, definitions, Finite automaton model, Acceptance of strings and languages, Deterministic finite automaton (DFA) and Non-deterministic finite automaton (NFA), Transitiondiagrams and Language recognizers. (Proofs Not Required)

Unit-II

Finite Automata: Acceptance of languages, Equivalence of NFA and DFA, NFA to DFA conversion, NFA with Ɛ- transitions, Significance, Conversion of NFA with Ɛ- transitions to NFA without Ɛ- transitions, Minimization of finite automata, Equivalence between two DFA's, Finite automata with output - Moore and Mealy machines, Equivalence between Moore and Mealy machines, conversion of Moore to Mealy and Mealy to Moore. (Proofs Not Required)

Unit-III

Regular Languages: Regular sets, Regular expressions, Operations and applications of regular expressions, Identity rules, Conversion of a given regular expression into a finite automaton, Conversion of finite automata into a regular expression.

Grammar Formalism: Regular grammars, Right linear and left linear grammars, Conversion from left linear to right linear grammars, Equivalence of regular grammar and finite automata, Inter conversion. (Proofs Not Required)

Unit- IV

Context Free Grammars: Context free grammars and languages, Derivation trees, Leftmost and Rightmost derivation of strings and sentential forms Ambiguity, left recursion and left factoring in context free grammars, Minimization of context free grammars, Normal forms for context free grammars, Chomsky normal form, Greibach normal form. (Proofs Not Required)

Unit-V

Pushdown Automata: Pushdown automata, definition, model, Graphical notation, Instantaneous descriptions, Acceptance of context free languages, Acceptance by final state

and acceptance by empty state and its equivalence, Equivalence of context free grammars and pushdown automata, Inter-conversion. Definition of Context Sensitive Grammar (CFG) and Linear bounded automata (LBA)**.** (Proofs Not Required)

Unit-VI

Turing Machine: Chomsky hierarchy on Languages, Turing Machine, definition, model, Instantaneous descriptions, Representation of Turing machines, Design of Turing machines, Types of Turing machines, Decidable and un-decidable problems, Definition of classes P and NP problems, NP complete and NP hard problems. (Proofs Not Required)

Text Books:

1. Introduction to Automata Theory Languages & Computation, 3/e, Hopcroft, Ullman, PEA
2. Introduction to Theory of Computation, 2/e, Sipser, Thomson

Reference Books:

1. Theory of Computation, Rajesh Shukla, Cengage, 2010
2. Theory of Computer Science, Automata languages and computation, 2/e, Mishra, Chandra shekaran, PHI
3. Introduction to Computer Theory, Daniel I.A. Cohen, John Wiley.

II Year B.Tech. (CSE). – II Semester

Computer Organization

Unit-I

Objectives: Gives a view of computer system from user's perspective, representation of data

Basic Structure of Computers: Computer Types, Functional unit, Basic Operational concepts, Bus structures

Data Representation: Data types, Complements, Fixed Point Representation. Floating– Point Representation, Other Binary Codes, Error Detection codes.

Unit-II

Objectives: Understanding RTL, Micro operations, ALU, Organization of stored program computer, types of instructions and design of basic components of the system

Register Transfer Language And Micro operations: Register Transfer language. Register Transfer Bus and memory transfers, Arithmetic Micro operations, logic micro operations, shift micro operations, Arithmetic logic shift unit.

Basic Computer Organization and Design: Instruction codes, Computer Register Computer instructions, Timing and control, Instruction cycle, Memory – Reference Instructions. Input – Output and Interrupt, Design of basic computer, Design of Accumulator Logic.

Unit-III

Objectives: Illustration of data paths and control flow for sequencing in CPUs, Microprogramming of control unit of CPU

Central Processing Unit: General Register Organization, STACK organization, Instruction formats. Addressing modes, DATA Transfer and manipulation, Program control, Reduced Instruction set computer.

Micro Programmed Control: Control memory, Address sequencing, micro program example, design of control unit

Unit-IV

Objectives: Illustration of algorithms for basic arithmetic operations using binary and decimal representation

Computer Arithmetic: Addition and subtraction, multiplication Algorithms, Division Algorithms, Floating – point Arithmetic operations, Decimal Arithmetic unit, Decimal Arithmetic operations.

Unit-V

Objectives: Description of different parameters of a memory system, organization and mapping of various types of memories

The Memory System: Memory Hierarchy, Main memory, Auxiliary memory, Associative Memory, Cache Memory, Virtual Memory.

Unit-VI

Objectives: Describes the means of interaction devices with CPU, their characteristics, modes and introduction multiprocessors.

Input-Output Organization: Peripheral Devices, Input-Output Interface, Asynchronous data transfer, Modes of Transfer, Priority Interrupts, Direct memory Access.

Multi Processors: Introduction, Characteristics or Multiprocessors, Interconnection Structures, Inter processor Arbitration.

Text Books:

1. Computer System Architecture, M.Moris Mano, 3rd Edition, Pearson/PHI
2. Computer Organization, Carl Hamacher, Zvonks Vranesic, SafeaZaky, 5th Edition, McGraw Hill.

References:

1. Computer Organization and Architecture – William Stallings Sixth Edition, Pearson/PHI
2. Structured Computer Organization – Andrew S. Tanenbaum, 4th Edition PHI/Pearson
3. Fundamentals or Computer Organization and Design, - Sivaraama Dandamudi Springer Int. Edition.

II Year B.Tech. (CSE)– II Semester

Advanced Data Structures Lab

1. To implement functions of Dictionary using Hashing (division method, Multiplication method, Universal hashing)
2. To perform various operations i.e, insertions and deletions on AVL trees
3. To perform various operations i.e., insertions and deletions on 2-3 trees.
4. To implement operations on binary heap.
5. To implement operations on graphs
    a) vertex insertion
    b) Vertex deletion
    c) finding vertex
    d) Edge addition and deletion
6. To implement Depth First Search for a graph non recursively.
7. To implement Breadth First Search for a graph non recursively.
8. To implement Prim's algorithm to generate a min-cost spanning tree.
9. To implement Krushkal's algorithm to generate a min-cost spanning tree.
10. To implement Dijkstra's algorithm to find shortest path in the graph.
11. To implement pattern matching using Boyer-Moore algorithm.
12. To implement Knuth-Morris-Pratt algorithm for pattern matching.

II Year B.Tech. (CSE)– II Semester

Java Programming Lab

1. Write a JAVA program to display default value of all primitive data types of JAVA
2. Write a JAVA program that displays the roots of a quadratic equation ax2+bx+c=0. Calculate the discriminent D and basing on the value of D, describe the nature of roots.
3. Write a JAVA program to display the Fibonacci sequence
4. Write a JAVA program give example for command line arguments.
5. Write a JAVA program to sort given list of numbers.
6. Write a JAVA program to search for an element in a given list of elements (linear search).
7. Write a JAVA program to search for an element in a given list of elements using binary search mechanism.
8. Write a JAVA program to determine the addition of two matrices.

9. Write a JAVA program to determine multiplication of two matrices.
10. Write a JAVA program to sort an array of strings
11. Write a JAVA program to check whether given string is palindrome or not.
12. Write a JAVA program for the following
    a. Example for call by value.
    b. Example for call by reference.
13. Write a JAVA program to give the example for 'this' operator. And also use the'this' keyword as return statement.
14. Write a JAVA program to give the example for 'super' keyword.
15. Write a JAVA program to demonstrate static variables, methods, and blocks.
16. Write a JAVA program that illustrates simple inheritance.
17. Write a JAVA program that illustrates multi-level inheritance
18. Write a JAVA program demonstrating the difference between method overloading and method overriding.
19. Write a JAVA program demonstrating the difference between method overloading and constructor overloading.
20. Write a JAVA program that describes exception handling mechanism.
21. Write a JAVA program for example of try and catch block. In this check whether the given array size is negative or not.
22. Write a JAVA program to illustrate sub class exception precedence over base class.
23. Write a JAVA program for creation of user defined exception.
24. Write a JAVA program to illustrate creation of threads using runnable class.(start method start each of the newly created thread. Inside the run method there is sleep() for suspend the thread for 500 milliseconds).
25. Write a JAVA program to create a class MyThread in this class a constructor, call the base class constructor, using super and starts the thread. The run method of the class starts after this. It can be observed that both main thread and created child thread are executed concurrently
26. Write a JAVA program illustrating multiple inheritance using interfaces.
27. Write a JAVA program to create a package named pl, and implement this package in ex1 class.
28. Write a JAVA program to create a package named mypack and import it in circle class.
29. Write a JAVA program to give a simple example for abstract class.
30. Write a JAVA program that describes the life cycle of an applet.
    a. Write a JAVA program to create a dialogbox and menu.
    b. Write a JAVA program to create a grid layout control.
31. Write a JAVA program to create a border layout control.
32. Write a JAVA program to create a padding layout control
33. Write a JAVA program to create a simple calculator.
34. Write a JAVA program that displays the x and y position of the cursor movement using Mouse.
35. Write a JAVA program that displays number of characters, lines and words in a text file

II Year B.Tech. (CSE) – II Semester

Free Open Source Software (Foss) Lab

1. **Session-1**
   - a Log into the system
   - b Use vi editor to create a file called myfile.txt which contains some text.
   - c Correct typing errors during creation.
   - d Save the file
   - e logout of the system

   Session-2
   - a Log into the system
   - b open the file created in session 1
   - c Add some text
   - d Change some text
   - e Delete some text
   - f Save the Changes
   - g Logout of the system

2.
   - a Log into the system
   - b Use the cat command to create a file containing the following data. Call it mytable use tabs to separate the fields.

1425	Ravi	15.65
4320	Ramu	26.27
6830	Sita	36.15
1450	Raju	21.86

   - c Use the cat command to display the file, mytable.
   - d Use the vi command to correct any errors in the file, mytable.
   - e Use the sort command to sort the file mytable according to the first field. Call the sorted file my table (same name)
   - f Print the file mytable
   - g Use the cut and paste commands to swap fields 2 and 3 of mytable. Call it my table (same name)
   - h Print the new file, mytable
   - i Logout of the system.

3. **I.**
   - a Login to the system
   - b Use the appropriate command to determine your login shell
   - c Use the /etc/passwd file to verify the result of step b.
   - d Use the who command and redirect the result to a file called myfile1. Use the more command to see the contents of myfile1.
   - e Use the date and who commands in sequence (in one line) such that the output of date will display on the screen and the output of who will be redirected to a file called myfile2. Use the more command to check the contents of myfile2.

II.

a Write a sed command that deletes the first character in each line in a file.

b Write a sed command that deletes the character before the last character in each line in a file

c Write a sed command that swaps the first and second words in each line in a file.

4.

a Pipe your /etc/passwd file to awk, and print out the home directory of each user.

b Develop an interactive grep script that asks for a word and a file name and then tells how many lines contain that word.

c Repeat

d Part using awk

5.

a Write a shell script that takes a command –line argument and reports on whether it is directory, a file, or something else.

b Write a shell script that accepts one or more file name as arguments and converts all of them to uppercase, provided they exist in the current directory.

c Write a shell script that determines the period for which a specified user is working on the system.

6.

a Write a shell script that accepts a file name starting and ending line numbers as arguments and displays all the lines between the given line numbers.

b Write a shell script that deletes all lines containing a specified word in one or more files supplied as arguments to it.

7.

a Write a shell script that computes the gross salary of a employee according to the following rules:

i. If basic salary is < 1500 then HRA =10% of the basic and DA =90% of the basic.

ii. If basic salary is >=1500 then HRA =Rs500 and DA=98% of the basic

ii. The basic salary is entered interactively through the key board.

b Write a shell script that accepts two integers as its arguments and computers the value of first number raised to the power of the second number.

8.

a Write an interactive file-handling shell program. Let it offer the user the choice of copying, removing, renaming, or linking files. Once the user has made a choice, have the program ask the user for the necessary information, such as the file name, new name and so on.

b Write shell script that takes a login name as command – line argument and reports when that person logs in

c Write a shell script which receives two file names as arguments. It should check whether the two file contents are same or not. If they are same then second file

should be deleted.

9.
   a. Write a shell script that displays a list of all the files in the current directory to which the user has read, write and execute permissions.
   b. Develop an interactive script that ask for a word and a file name and then tells how many times that word occurred in the file.
   c. Write a shell script to perform the following string operations:
      i. To extract a sub-string from a given string.
      ii. To find the length of a given string.

10. Write a C program that takes one or more file or directory names as command line input and reports the following information on the file:
    i. File type
    ii. Number of links
    iii. Read, write and execute permissions
    iv. Time of last access
    (Note : Use stat/fstat system calls)

11. Write C programs that simulate the following unix commands:
    a)mv b)cp (Use system calls)

12. Write a C program that simulates ls Command (Use system calls / directory API)

13. Do the following Shell programs also
    1) Write a shell script to check whether a particular user has logged in or not. If he has logged in, also check whether he has eligibility to receive a message or not
    2) Write a shell script to accept the name of the file from standard input and perform the following tests on it
       a. File executable
       b. File readable
       c. File writable
       d. Both readable & writable
    3) Write a shell script which will display the username and terminal name who login recently in to the unix system
    4) Write a shell script to find no. of files in a directory
    5) Write a shell script to check whether a given number is perfect or not
    6) Write a menu driven shell script to copy, edit, rename and delete a file
    7) Write a shell script for concatenation of two strings
    8) Write a shell script which will display Fibonacci series up to a given number of argument
    9) Write a shell script to accept student number, name, marks in 5 subjects. Find total, average and grade. Display the result of student and store in a file called stu.dat
       Rules: avg>=80 then grade A
       Avg<80&&Avg>=70 then grade B
       Avg<70&&Avg>=60 then grade C
       Avg<60&&Avg>=50 then grade D
       Avg<50&&Avg>=40 then grade E Else grade F
    10) Write a shell script to accept empno,empname,basic. Find DA,HRA,TA,PF

using following rules. Display empno, empname, basic, DA,HRA,PF,TA,GROSS SAL and NETSAL. Also store all details in a file called emp.dat

Rules: HRA is 18% of basic if basic > 5000 otherwise
550 DA is 35% of basic
PF is 13% of basic
IT is 14% of basic
TA is 10% of basic

11) Write a shell script to demonstrate break and continue statements
12) Write a shell script to satisfy the following menu options
    a. Display current directory path
    b. Display todays date
    c. Display users who are connected to the unix system
    d. Quit
13) Write a shell script to delete all files whose size is zero bytes from current directory
14) Write a shell script to display string palindrome from given arguments
15) Write a shell script which will display Armstrong numbers from given arguments
16) Write a shell script to display reverse numbers from given argument list
17) Write a shell script to display factorial value from given argument list
18) Write a shell script which will find maximum file size in the given argument list
19) Write a shell script which will greet you "Good Morning", "Good Afternoon", "Good Evening' and "Good Night" according to current time
20) Write a shell script to sort the elements in a array using bubble sort technique
21) Write a shell script to find largest element in a array
22) Write an awk program to print sum, avg of students marks list
23) Write an awk program to display students pass/fail report
24) Write an awk program to count the no. of vowels in a given file
25) Write an awk program which will find maximum word and its length in the given input File
26) Write a shell script to generate the mathematical tables.
27) Write a shell script to sort elements of given array by using selection sort.
28) Write a shell script to search given number using binary search.
29) Write a shell script to find number of vowels, consonants, numbers, white spaces and special characters in a given string.
30) Write a shell script to lock the terminal.