II Year - I Semester

	II- I SEMESTER		L	T	P	C
1-BS	**16BH3T07**	Statistics with R Programming	4	-	-	3
2	**16CS3T02**	Mathematical Foundations of Computer Science	4	-	-	3
3	**16EC3T08**	Digital Logic Design	4	-	-	3
4	**16IT3T02**	Python Programming	4	-	-	3
5	**16CS3T03**	Data Structures through C++	4	-	-	3
6	**16BH3T14**	Managerial Economics and Financial Analysis	4	-	-	3
7	**16CS3L02**	Data Structures through C++Lab	-	-	3	2
8	**16IT3L02**	Python Programming Lab	-	-	3	2
		Total Credits				**22**

II Year - II Semester

	II- II SEMESTER		L	T	P	C
1	**16CS4T04**	Software Engineering	4	-	-	3
2	**16IT4T06**	Java Programming	4	-	-	3
3	**16CS4T05**	Advanced Data Structures	4	-	-	3
4	**16CS4T06**	Computer Organization	4	-	-	3
5	**16CS4T07**	Formal Languages and Automata Theory	4	-	-	3
6	**16CS4T08**	Computer Graphics	4	-	-	3
7	**16CS4L03**	Advanced Data Structures Lab	-	-	3	2
8	**16IT4L04**	Java Programming Lab	-	-	3	2
		Total Credits				**22**

SYLLABUS

II Year - I Semester

L	T	P	C
4	1	0	3

STATISTICS WITH R PROGRAMMING

Learning Objectives:

1. The course is designed to equip the students with the necessary statistical skills and techniques that are essential part of R programming.
2. The skills derived from the course will help the student how to write the R program and interpret the statistics to analyze and make decision to the data.

Course Outcomes: At the end of the Course, Student will be able to:

1. Install the R software and writing the R programming for given statistical data
2. Interpret the statistical data analysis with R programming

UNIT I: Discrete probability distributions and Introduction to R

Descriptive Statistics - Random variables - Discrete random variable - Expectation - Binomial, Poisson distributions.

Introduction to R software - Vectors - Matrices - Arrays - Lists - Data frames - Basic arithmetic operations in R - Importing and exporting files in R.

UNIT II: Continuous Probability distribution and Computing with R

Continuous random variable - Normal distribution - Properties - Gamma distribution - Weibul distribution.

R commands for computing probability distributions.

UNIT III: Sampling Theory and Test of Hypothesis

Sampling - Central limit theorem (without proof) - Sampling distribution of means - point estimation - interval estimation

Construction of confidence intervals using R.

UNIT IV – Test of Significance

Introduction to test of Hypothesis - Type-I Error - Type-II Error - One tail and Two tail tests concerning single mean and two means - single proportion - two proportions.

R programming for Z-test, t-test and F-test and Chi square test.

UNIT V: Analysis of Variance

ANOVA for one way classification - ANOVA for two way classification.

R programming – ANOVA for one way classification – ANOVA for two way classification.

UNIT VI: Correlation and regression

Simple correlation and regression – Regression by the method of least squares – Rank correlation – Multiple linear regression

R programming for correlation and regression.

Text Books:

1. **Miller and John E. Freund,** Probability and Statistics for Engineers, Prentice Hall of India.
2. **G. Jay Kerns**, Introduction to Probability and Statistics Using R, First Edition ISBN: 978-0-557-24979-4. (Free e-book from R software website)

Reference Books:

1. **Robert I. Kabacoff,** R in Action, Second Edition, Data analysis and graphics with R ISBN: 9781617291388, Printed in the United States of America.
2. **Erwin Kreyszig**, Advanced Engineering Mathematics, 10th Edition, Wiley-India
3. **T.K.V. Iyengar et al.,** Probability and Statistics, S Chand Publications.
4. **Jay L. Devore,** Probability and Statistics for Engineering and Sciences, 8th Edition, Cengage Learning. ISBN 13: 978-81-315-1839-7.

PRAGATI ENGINEERING COLLEGE
(Autonomous)
1-378, ADB Road, Surampalem, E.G.District, A.P. - 533 437
(Approved by AICTE & Permanently Affiliated to JNTUK, Kakinada & Accredited by NAAC with 'A' Grade)
(Recognised by UGC Under Sections 2 (f) and 12 (b) of UGC act, 1956)
Ph : 08852 - 252233, 252234, 252235, Fax : 252232, Website : www.pragati.ac.in
(Sponsored by Gayatri Educational Society)
D.No. 2-24-4/2, Ground Floor, Janmabhoomi Park Road, Srinagar, Kakinada - 3, Ph : 0884 - 2355900, Fax : 2363900

Department of Computer Science and Engineering

SYLLABUS

II Year - I Semester

L	T	P	C
4	1	0	3

MATHEMATICAL FOUNDATIONS OF COMPUTER SCIENCE
(Common to CSE, IT)

Learning objectives:
Students are expected to learn:
1. The syntax and semantics of propositional and predicate logic.
2. How basic concepts in Algebra can be applied in computer science.
3. Proof techniques such as Mathematical Induction and Contradiction, these techniques will come in handy for courses such as Analysis of Algorithms and Automata Theory.
4. Understanding of Number Theory will help in Cryptanalysis.
5. To explain with examples the basic terminology of functions, relations, and sets.
6. To perform the operations associated with sets, functions, and relations.
7. To use Graph Theory for solving problems

Course Outcomes:
Upon successful completion of this course, students will be able to:
1. Learn the Difference between Propositional Logic and Predicate Logic.
2. Solve the problems related to fundamental theorems.
3. Paraphrase the properties of homomorphism and POSET and Lattices and Apply induction and other proof techniques towards solving recurrences and other problems in elementary algebra.
4. Ability to Illustrate by examples the basic terminology of functions, relations, and sets and demonstrate knowledge of their associated operations.
5. Ability to demonstrate in practical applications the use of basic counting principles of permutations, combinations, inclusion/exclusion principle and the pigeonhole methodology.
6. Ability to represent and Apply Graph theory in solving computer science problems.

UNIT - I
Mathematical logic:
Propositional calculus: statements and notations, connectives, Truth tables, Tautologies, Equivalence of formulas, Tautological implications, Normal forms, Theory of inference for statement calculus.
Predicate Calculus: predicate logic, statement functions, variables and quantifiers, free and bound variables.

UNIT - II

Number Theory:

Properties of integers, Division Theorem, The greatest Common Divisor, Euclidean Algorithm, Least Common Multiple, Testing Prime numbers, The Fundamental Theorem of Arithmetic (Fermat's Theorem and Euler's Theorem) Mathematical induction– Principle of Mathematical Induction, Exercises.

UNIT - III

Relations: Properties of Binary Relations, equivalence, transitive closure, compatibility and partial ordering relations, Lattices, Hasse diagram.

Functions: Inverse Function Composition of functions, recursive Functions, Lattice and its Properties

UNIT - IV

Graph Theory :

Basic Concepts of Graphs, Matrix representation of graphs: Adjacency Matrices, Incidence Matrices, Isomorphic Graphs, Paths and Circuits, Eulerian graphs, Planar Graphs, Graph coloring, spanning trees.

UNIT - V

Algebraic Structures:

Algebraic systems – Semi groups and monoids, Homomorphism of Semi group and Monoids, Groups, Cosets. Partial ordering – Posets – Lattices as Posets

Elementary Combinatorics:

Basis of counting, Combinations & Permutations, with repetitions, Binomial Coefficients, Binomial Multinomial theorems, the principles of
Inclusion – Exclusion. Pigeon hole principles and its application.

UNIT - VI

Recurrence Relations:

Generating Function of Sequences, Partial Fractions, Calculating coefficient of Generating Functions recurrence relations. Formulation as Recurrence relations, solving linear recurrence relations, methods of Characteristics roots, solutions of Inhomogeneous recurrence relations

Text Books

1. Discrete Mathematics for Computer Scientists & Mathematicians, 2/e Mott, Kandel, Baker, PHI.
2. Trembly J.P. and Manohar R, "Discrete Mathematical Structures with Applications to Computer Science", Tata McGraw–Hill Pub. Co. Ltd, New Delhi, 30th Re-print (2007).
3. Kenneth H.Rosen, "Discrete Mathematics and its Applications", 6th Edition, Special Indian edition, Tata McGraw – Hill Pub. Co. Ltd., New Delhi, (2007).

Reference Books

1. Ralph. P. Grimaldi, "Discrete and Combinatorial Mathematics: An Applied Introduction", Fourth Edition, Pearson Education Asia, Delhi, (2002).
2. Thomas Koshy, "Discrete Mathematics with Applications", Elsevier Publications, (2006).
3. Seymour Lipschutz and Mark Lipson, "Discrete Mathematics", Schaum's Outlines, Tata McGraw – Hill Pub. Co. Ltd., New Delhi, 2007,Second edition, Fifth reprint.

URLs

1. https://www.cs.cmu.edu/~emc/15414-f12/lecture/propositional_logic.pdf.
2. https://www.tutorialspoint.com/discrete_mathematics/discrete_mathematics_propositional_logic.htm
3. https://www.tutorialspoint.com/discrete_mathematics/discrete_mathematics_functions.htm
4. http://discretemathnotes.blogspot.in/2008/08/groups.html
5. https://people.cs.pitt.edu/~milos/courses/cs441/lectures/Class25.pdf

SYLLABUS

II Year - I Semester

L	T	P	C
4	1	0	3

DIGITAL LOGIC DESIGN

Learning Objectives:

1. To solve a typical number base conversion and analyze new error coding techniques.
2. Theorems and functions of Boolean algebra and behavior of logic gates.
3. To optimize logic gates for digital circuits using various techniques.
4. Boolean function simplification using Karnaugh maps and Quine-McCluskey methods.
5. To understand concepts of combinational circuits.
6. To develop advanced sequential circuits.

Course Outcomes:

1. Classify different number systems and apply to generate various codes.
2. Use the concept of Boolean algebra in minimization of switching functions
3. Design different types of combinational logic circuits.
4. Apply knowledge of flip-flops in designing of Registers and counters
5. The operation and design methodology for synchronous sequential circuits and algorithmic state machines.
6. Produce innovative designs by modifying the traditional design techniques.

UNIT I

Number Systems: Binary, Octal, Decimal, Hexadecimal Number Systems. Conversion of Numbers from One Radix To Another Radix, r's Complement and (r-1)'s Complement Subtraction of Unsigned Numbers, Problems, Signed Binary Numbers, Weighted and Non weighted codes

UNIT II

Logic Gates And Boolean Algebra: Basic Gates NOT, AND, OR, Boolean Theorms, Complement And Dual of Logical Expressions, UniversalGates, Ex-Or and Ex-Nor Gates, SOP,POS, Minimizations of Logic Functions Using Boolean Theorems,Two level Realization of Logic Functions Using Universal Gates

Gate Level Minimization: Karnaugh Map Method (K-Map): Minimization of Boolean Functions maximumupto Four Variables, POS and SOP, Simplifications With Don't Care Conditions Using K-Map.

UNIT III

Combinational Logic Circuits: Design of Half Adder, Full Adder, Half Subtractor, Full Subtractor, Ripple Adders and Subtractors, Ripple Adder/Subtractor Using Ones and Twos Complement Method.Design of Decoders, Encoders, Multiplexers, Demultiplexers, Higher Order Demultiplexers and Multiplexers, Priority Encoder, Code Converters,Magnitude Comparator.

UNIT IV

Introduction to Sequential Logic Circuits: Classification of Sequential Circuits, Basic Sequential Logic Circuits: Latch and Flip-Flop, RS- Latch UsingNAND and NOR Gates, Truth Tables. RS,JK,T and D Flip Flops, Truth and Excitation Tables, Conversionof Flip Flops. Flip Flops With Asynchronous Inputs (Preset and Clear).

UNIT V

Registers and Counters: Design of Registers, Buffer Register, Control Buffer Registers, Bidirectional Shift Registers, Universal ShiftRegister, Design of Ripple Counters, Synchronous Counters and Variable Modulus Counters, Ring Counter,Johnson Counter.

UNIT VI

Introduction to Programmable Logic Devices **(PLDs)**: PLA, PAL, PROM. Realization of Switching Functions Using PROM, PAL and PLA. Comparison of PLA, PAL and PROM.

TEXT BOOKS :

1. Digital Design ,4/e, M.Morris Mano, Michael D Ciletti, PEA
2. Switching and Finite Automata Theory, 3/e, Kohavi, Jha, Cambridge.

REFERENCE BOOKS

1. Fundamentals of Logic Design, 5/e, Roth, Cengage
2. Digital Logic Design, Leach, Malvino, Saha,TMH
3. Modern Digital Electronics, R.P. Jain, TMH

Web Links :

1. https://www.youtube.com/watch?v=CeD2L6KbtVM
2. Lecture series on Digital Circuits & Systems by Prof.S.Srinivasan, Department of Electrical Engineering, IIT Madras.For more details on NPTEL visit http://nptel.iitm.ac.in
3. https://www.youtube.com/watch?v=K73N9ES_8nI
4. https://www.youtube.com/watch?v=62WxkICo2Bc

SYLLABUS

II Year - I Semester

L	T	P	C
4	1	0	3

PYTHON PROGRAMMING
(Common to CSE, IT)

Learning objectives:

1.To make the students understand the fundamentals of python programming.
2. To expose the students to object oriented concepts.
3. To make the students to develop applications using python.
4. To make students to use python for automation.

Course Outcomes:

1.Understand various data types, operators in Python.
2.Write programs using loop and branch statements to manipulate data in files.
3.Perform string manipulations.
4.Perform list and dictionaries operations in python
5.Understand the concept of modular programming using functions.
6.Realize the ease of developing complex programs with smaller sized programs.

UNIT – I

Introduction to Python, Installing Python:

Basic syntax, interactive shell, editing, saving, and running a script. The concept of data types variables, assignments immutable variables numerical types; arithmetic operators and expressions comments in the program understanding error messages

UNIT – II

Conditions, Boolean logic, logical operators; ranges:

Control statements:

If-else, loops (for, while) short-circuit (lazy) evaluation, Strings and text files manipulating files and directories, os and sys modules; text files: reading/writing text and numbers from/to a filecreating and reading a formatted file (csv or tab-separated).

UNIT – III

String manipulations:

Subscript operator, indexing, slicing a string; strings and number system: converting strings to numbers and vice versa. Binary, octal, hexadecimal numbers, Lists, tuples, and dictionaries.

UNIT – IV

Basic list operators, replacing, inserting, removing an element; searching and sorting lists; dictionary literals, adding and removing keys, accessing and replacing values; traversing dictionaries.

UNIT – V

Design with functions: Hiding redundancy, complexity; arguments and return values; formal vs. Actual arguments, named arguments, Program structure and design, Recursive functions.

UNIT VI

Classes and OOP:

Classes, objects, attributes and methods; defining classes; design with classes, data modeling; persistent storage of objects, inheritance, polymorphism, operator overloading (eq_, _str_, etc) abstract classes; exception handling, try block, Multithreading, Automation using Python.

Text Books

1. Think Python, How to Think Like a Computer Scientist, Version 2.0.17, Allen Downey, Green Tea Press.

Reference Books

1. Python Essential Reference, David M. Beazley , Pearson Education, Inc.
2. Fluent Python, Luciano Ramalho by O'Reilly Media
3. Python Cookbook, David Beazley and Brian K. Jones, O'Reilly Atlas.3e
4. Fundamentals of Python: First Programs, Kenneth Lambert, Course Technology, Cengage Learning, 2012. ISBN-13: 978-1-111-82270-5.

URLs

1. Think Python: How to Think Like a Computer Scientist by Allen B. Downey http://www.greenteapress.com/thinkpython/thinkpython.html
2. Dive into Python by Mark Pilgrim- http://www.diveintopython.net http://staff.washington.edu/jon/python-course/
3. https://wiki.python.org/moin/PythonBooks

SYLLABUS

II Year - I Semester

L	T	P	C
4	1	0	3

DATA STRUCTURES THROUGH C++

(Common to CSE, IT)

Learning objectives:

1. Assess how the choice of data structures and algorithm design methods impacts the performance of programs.
2. Choose the appropriate data structure and algorithm design method for a specified application.
3. Solve problems using data structures such as linear lists, stacks, queues, binary trees, heaps binary search trees, and graphs and writing programs for these solutions.

Course Outcomes:

1. Distinguish between procedures and object oriented programming.
2. Apply advanced data structure strategies for exploring complex data structures.
3. Compare and contrast various data structures and design techniques in the area of performance.
4. Implement data structure algorithms through C++.
5. Incorporate data structures into the applications such as binary search trees, AVL and B Trees
6. Implement all data structures like stacks, queues, trees, lists and graphs and compare their Performance and trade offs

UNIT-I: ARRAYS

Abstract Data Types and the C++ Class, An Introduction to C++ Class- Data Abstraction and Encapsulation in C++- Declaring Class Objects and Invoking Member Functions- Special Class Operations- Miscellaneous Topics- ADTs and C++Classes, The Array as an Abstract Data Type, The Polynomial Abstract Data type- Polynomial Representation- Polynomial Addition. Spares Matrices, Introduction- Sparse Matrix Representation- Transposing a Matrix- Matrix Multiplication, Representation of Arrays.

UNIT-II: STACKS AND QUEUES

Templates in C++, Template Functions- Using Templates to Represent Container Classes, The Stack Abstract Data Type, The Queue Abstract Data Type, Subtyping and Inheritance in C++, Evaluation of Expressions, Expression- Postfix Notation- Infix to Postfix.

UNIT-III: LINKED LISTS

Single Linked List and Chains, Representing Chains in C++, Defining a Node in C++- Designing a Chain Class in C++- Pointer manipulation in C++- Chain Manipulation Operations, The Template Class Chain, Implementing Chains with Templates- Chain Iterators- Chain Operations- Reusing a Class, Circular Lists, Available Space Lists, Linked Stacks and Queues, Polynomials, Polynomial Representation- Adding Polynomials- Circular List Representation of Polynomials, Equivalence Classes, Sparse Matrices, Sparse Matrix Representation- Sparse Matrix Input- Deleting a Sparse Matrix, Doubly Linked Lists,

UNIT-IV: TREES

Introduction, Terminology, Representation of Trees, Binary Trees, The Abstract Data Type, Properties of Binary Tress, Binary Tree Representations, Binary Tree Traversal and Tree Iterators, Introduction, Inorder Traversal Preorder Traversal, Postorder Traversal, Thread Binary Trees, Threads, Inorder Traversal of a Threaded Binary Tree, Inserting a Node into a Threaded Binary Tree, , Binary Search Trees, Definition, Searching a Binary Search Tree, Insertion into a Binary Search Tree, Deletion from a Binary Search Tree, Height of Binary Search Tree.

UNIT-V: GRAPHS

The Graph Abstract Data Type, Introduction, Definition, Graph Representation, Elementary Graph Operation, Depth First Search, Breadth First Search,

UNIT-VI: SORTING

Insertion Sort, Quick Sort, Merge Sort Merging, Iterative Merge Sort, Recursive Merge Sort, Heap Sort.

TEXT BOOKS:

1. Fundamentals of Data structures in C++, E. Horowitz, S.Sahni, Dinesh Mehta, Galgotia Publications, 2006
2. Data structures and Algorithm Analysis in C++, Mark Allen Weiss, Pearson Education. Ltd., Second Edition.
3. Data structures and Algorithms in C++, Michael T.Goodrich, R.Tamassia and .Mount, Wiley student edition, John Wiley and Sons.

REFERENCE BOOKS:

1. Data structures and algorithms in C++, 3rd Edition, Adam Drozdek, Thomson
2. Data structures using C and C++, Langsam, Augenstein and Tanenbaum, PHI.
3. Problem solving with C++, The OOP, Fourth edition, W.Savitch, Pearson education.

URLs

1. https://www.tutorialspoint.com/cplusplus/cpp_data_structures.htm
2. http://people.cs.vt.edu/~shaffer/Book/C++3elatest.pdf
3. http://cds.iisc.ac.in/courses/ds286/
4. http://www.geeksforgeeks.org/data-structures

SYLLABUS

II Year - I Semester

L	T	P	C
4	1	0	3

MANAGERIAL ECONOMICS AND FINANCIAL ANALYSIS

(Common to all branches of Engineering)

Learning objectives:

1. To understand the concept and nature of Managerial Economic s and its relationship with other disciplines, Concept of Demand and Demand forecasting.
2. To understand the concept of Production function, Input Output relationship, different Cost Concepts and Concept of Cost-Volume-Profit Analysis.
3. To understand the Nature of Competition, Characteristics of Pricing in the different market structure and significance of various pricing methods.
4. To know the different forms of Business organization and their Merits and Demerits both public & private Enterprises and the concepts of Business Cycles.
5. To understand the different Accounting Systems preparation of Financial Statements.
6. To understand the concept of Capital, Capitalization, and Capital Budgeting and to know the techniques used to evaluate Capital Budgeting proposals by using different methods and uses of different tools for performance evaluation.

Course Outcomes:

1. The Learner is equipped with the knowledge of estimating the Demand for a product and the relationship between Price and Demand.
2. One should understand the Cost Concepts for decision making and to estimate the least cost combination of inputs.
3. One has to understand the nature of different markets and Price Output determination under various market conditions.
4. One should be equipped with the knowledge of different Business Units.
5. The Learner is able to prepare Financial Statements
6. The Learner is able to understand the usage of various Ratios for financial Analysis and evaluates various investment project proposals with the help of capital budgeting techniques for decision making.

Unit – I

Introduction to Managerial Economics and demand Analysis:

Definition of Managerial Economics and Scope-Managerial Economics and its relation with other subjects-Basic Economic Tools used in Managerial Economics-Concepts of Demand-Types-Determents-Law of Demand its Exception-Elasticity of Demand-Types and Measurement- Law of Supply -Demand forecasting and it's Methods.

Unit – II

Production and Cost Analyses:

Production function-Isoquants and Isocosts-Law of Variable proportions- Laws of Returns to Scale-Cobb-Douglas Production function-Economies of Scale-Cost Concepts- Fixed vs Variable Costs-Out of Pocket Costs vs Imputed Costs-Cost Volume Profit analysis-Determination of Break-Even Point (Simple Problems)

Unit – III

Introduction to Markets, Theories of the Firm & Pricing Policies:

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

Unit – IV

Types of Business Organization 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

Introduction to Accounting:

Introduction to Double Entry Systems-Journal-Ledger- Trail Balance - Preparation of Financial Statements - Analysis and Interpretation of Financial Statements-Ratio Analysis – liquidity ratios, profitability ratios, solvency ratios, turnover ratios
– Preparation of the Funds flow Statement (Simple Problems)

Unit – VI

Capital and Capital Budgeting: Capital Budgeting: Meaning of Capital-Capitalization-Sources of Finance (with special reference to Shares and Debentures)-Meaning of Capital Budgeting-Need for Capital Budgeting-Techniques of Capital Budgeting-Traditional and Modern Methods.

TEXT BOOKS

1. Dr. N. Appa Rao, Dr. P. Vijay Kumar: 'Managerial Economics and Financial Analysis', Cengage Publications, New Delhi – 2011

2. Dr. A. R. Aryasri – Managerial Economics and Financial Analysis, TMH 2011
3. Prof. J.V.Prabhakara rao, Prof. P. Venkatarao. 'Managerial Economics and Financial Analysis', Ravindra Publication.

REFERENCES

1. V. Maheswari: Managerial Economics, Sultan Chand.
2. Suma Damodaran: Managerial Economics, Oxford 2011.
3. Dr. B. Kuberudu and Dr. T. V. Ramana: Managerial Economics & Financial Analysis, Himalaya Publishing House 2011.
4. Vanitha Agarwal: Managerial Economics, Pearson Publications 2011.
5. Sanjay Dhameja: Financial Accounting for Managers, Pearson.
6. Maheswari: Financial Accounting, Vikas Publications.
7. S. A. Siddiqui & A. S. Siddiqui: Managerial Economics and Financial Analysis, New Age International Publishers, 2012

SYLLABUS

II Year - I Semester

L	T	P	C
0	0	3	2

DATA STRUCTURES THROUGH C++ LAB

(Common to CSE, IT)

Learning objectives:

This course is aimed to provide hands on experience to implement basic linear and nonlinear data structures. This course covers implementation of stack, queue, list, sorting techniques, binary search trees, and balanced search trees.

Course Outcomes:

At the end of this lab session, the student will

1. Be able to design and analyze the time and space efficiency of the data structure
2. Be capable to identity the appropriate data structure for given problem
3. Have practical knowledge on the application of data structures

List of Experiments

1. a) Write a Program to implement linear search algorithm.
   b) Write a Program to implement binary search algorithm.

2. Write a Program to Sort the set of elements by using
   i) Quick Sort. ii) Merge Sort.

3. a) Write a Program to Implement Stack Operations by using Array.
   b) Write a Program to implement the operations of Queue using array.

4. a) Write a program that uses stack operations to convert a given infix expression into its postfix.
   b) Write a program that uses stack operations to evaluate given postfix expression.

5. Write a Program to Implement Singly Linked List and its operations.

6. a) Write a Program to Implement Stack Operations by using Linked List.
   b) Write a Program to implement the operations of Queue using linked list.

7. Write a Program to Implement Circular Queue Operations by using Array.

8. Write a Program to Implement the Binary Search Tree Operations.

9. Write a Program to Perform the Tree Traversal Techniques by using Iterative Method

10. Write a Program to Perform the Tree Traversal Techniques by using recursion.

11. Write a program for implementing the Depth First Search graph traversal algorithm.

12. Write a program for implementing the Breadth First Search graph traversal algorithm.

Note: Use Classes and Objects to implement the above programs.

PRAGATI ENGINEERING COLLEGE

(Autonomous)

1-378, ADB Road, Surampalem, E.G.District, A.P. - 533 437

(Approved by AICTE & Permanently Affiliated to JNTUK, Kakinada & Accredited by NAAC with 'A' Grade)

(Recognised by UGC Under Sections 2 (f) and 12 (b) of UGC act, 1956)

Ph : 08852 - 252233, 252234, 252235, Fax : 252232, Website : www.pragati.ac.in

(Sponsored by Gayatri Educational Society)

D.No. 2-24-4/2, Ground Floor, Janmabhoomi Park Road, Srinagar, Kakinada - 3, Ph : 0884 - 2355900, Fax : 2363900

Department of Computer Science and Engineering

SYLLABUS

II Year - I Semester

L	T	P	C
0	0	3	2

PYTHON PROGRAMMING LAB

(Common to CSE& IT)

Learning objectives:

1.To practice the students understand the fundamentals of python programming.
2. To expose the students to object oriented concepts.
3. To practice the students to develop applications using python.
4. To practice students to use python for automation.

Course Outcomes:

1.Use various data types, operators in Python.
2.Write programs using loop and branch statements to manipulate data in files.
3.Perform string manipulations.
4.Perform list and dictionaries operations in python
5.Use functions for modular programming.

Exercise 1 - Basics

a)Running instructions in Interactive interpreter and a Python Script
b) Write a program to purposefully raise Indentation Error and Correct it

Exercise 2 - Operations

a)Write a program to compute distance between two points taking input from the user (Pythagorean Theorem)
b) Write a program to convert a given decimal number to other base systems

Exercise - 3 Control Flow

a) Write a program to calculate the Income Tax as per the rules of Indian Government.
b) Each new term in the Fibonacci sequence is generated by adding the previous two terms. By starting with 1 and 2, the first 10 terms will be:

1, 2, 3, 5, 8, 13, 21, 34, 55, 89

By considering the terms in the Fibonacci sequence whose values do not exceed four million, find the sum of the even-valued terms.

Exercise - 4 - DS

a) Write a program to count the numbers of characters in the string and store them in a dictionary data structure

b) Write a program to use split and join methods in the string and trace a birthday with a dictionary data structure.

Exercise - 5 Files

a) Write a program encrypts the message in a given file and write the encrypted message in an output file. Write a decrypt program and verify.

b) Write a program to compute the number of characters, words and lines in a file.

Exercise -6 Functions

a) Write a function ball collides that takes two balls as parameters and computes if they are colliding. Your function should return a Boolean representing whether or not the balls are colliding.

Hint: Represent a ball on a plane as a tuple of (x, y, r), r being the radius

If (distance between two balls centers) <= (sum of their radii) then (they are colliding)

b) Find mean, median, mode for the given set of numbers in a list.

Exercise -7 Functions - Continued

a) Write a function nearly equal to test whether two strings are nearly equal. Two strings a and b are nearly equal when a can be generated by a single mutation on b.

b) Write a function dups to find all duplicates in the list.

c) Write a function unique to find all the unique elements of a list.

Exercise 8 - Multi-D Lists

a) Write a program that defines a matrix and prints

b) Write a program to perform addition of two square matrices

c) Write a program to perform multiplication of two square matrices

Exercise - 9 - Modules

a) Install packages requests, flask and explore them. using (pip)
b) Write a script that imports requests and fetch content from the page. Eg. (Wiki)
c) Write a simple script that serves a simple HTTP Response and a simple HTML Page

Exercise - 10 OOP

a) Class variables and instance variable and illustration of the self-variable for ATM Machine

Exercise - 11 – Advanced

Write a program (using threads) to allocate 100 students (each student modeled as a single thread) to 5 sections of a course.

PRAGATI ENGINEERING COLLEGE
(Autonomous)
1-378, ADB Road, Surampalem, E.G.District, A.P. - 533 437
(Approved by AICTE & Permanently Affiliated to JNTUK, Kakinada & Accredited by NAAC with 'A' Grade)
(Recognised by UGC Under Sections 2 (f) and 12 (b) of UGC act, 1956)
Ph : 08852 - 252233, 252234, 252235, Fax : 252232, Website : www.pragati.ac.in
(Sponsored by Gayatri Educational Society)
D.No. 2-24-4/2, Ground Floor, Janmabhoomi Park Road, Srinagar, Kakinada - 3, Ph : 0884 - 2355900, Fax : 2363900

Department of Computer Science and Engineering

SYLLABUS

II Year - II Semester

L	T	P	C
4	1	0	3

SOFTWARE ENGINEERING
(Computer Science & Engineering)

Learning objectives:

To develop skills that will enable them to construct software of high quality – software that is reliable, and that is reasonably easy to understand, modify and maintain.

Course Outcomes:

1. Knowledge of basic SW engineering methods and practices, and their appropriate application;
2. An understanding of software requirements and the SRS document.
3. An understanding of different software architectural styles.
4. An understanding of implementation issues such as modularity and coding standards.
5. An understanding of software testing approaches
6. An understanding on quality control and how to ensure good quality software.

UNIT-I:

Software and Software Engineering: The Nature of Software, The Unique Nature of WebApps, Software Engineering, Software Process, Software Engineering Practice, Software Myths.

Process Models: A Generic Process Model, Process Assessment and Improvement, Prescriptive Process Models, Specialized Process Models, The Unified Process, Personal and Team Process Models, Process Terminology, Product and Process.

UNIT-II:

Requirements Analysis and Specification: Requirements Gathering and Analysis, Software Requirement Specification (SRS), Formal System Specification.

Software Design: Overview of the Design Process, How to Characterize of a Design? Cohesion and Coupling, Layered Arrangement of Modules, Approaches to Software Design

UNIT – III:

Function-Oriented Software Design: Overview of SA/SD Methodology, Structured Analysis, Developing the DFD Model of a System, Structured Design, Detailed Design, Design Review, over view of Object Oriented design.

User Interface Design: Characteristics of Good User Interface, Basic Concepts, Types of User Interfaces, Fundamentals of Component-based GUI Development, A User Interface Design Methodology.

UNIT – IV:

Coding And Testing: Coding, Code Review, Software Documentation, Testing, Unit Testing, Black-Box Testing, White-Box Testing, Debugging, Program Analysis Tool, Integration Testing, Testing Object-Oriented Programs, System Testing, Some General Issues Associated with Testing

UNIT – V:

Software Reliability And Quality Management: Software Reliability, Statistical Testing, Software Quality, Software Quality Management System, ISO 9000, SEI Capability Maturity Model.

Computer Aided Software Engineering: Case and its Scope, Case Environment, Case Support in Software Life Cycle, Other Characteristics of Case Tools, Towards Second Generation CASE Tool, Architecture of a Case Environment.

UNIT – VI

Software Maintenance: Software maintenance, Maintenance Process Models, Maintenance Cost, Software Configuration Management.

Software Reuse: what can be Reused? Why almost No Reuse So Far? Basic Issues in Reuse Approach, Reuse at Organization Level.

TEXT BOOKS:

1. Software Engineering A practitioner's Approach, Roger S. Pressman, Seventh Edition McGrawHill International Edition.
2. Fundamentals of Software Engineering, Rajib Mall, Third Edition, PHI.
3. Software Engineering, Ian Sommerville, Ninth edition, Pearson education

REFERENCE BOOKS:

1. Software Engineering : A Primer, Waman S Jawadekar, Tata McGraw-Hill, 2008
2. Software Engineering, A Precise Approach, Pankaj Jalote, Wiley India,2010.
3. Software Engineering, Principles and Practices, Deepak Jain, Oxford University Press.
4. Software Engineering1: Abstraction and modeling, Diner Bjorner, Springer International edition, 2006.

URLs

1. https://www.tutorialspoint.com/software_engineering/software_engineering_tutorial.pdf
2. http://nptel.ac.in/downloads/106105087/

SYLLABUS

II Year - II Semester

L	T	P	C
4	1	0	3

JAVA PROGRAMMING
(Common to CSE & IT)

Learning objectives:

1. To make the students understand the fundamentals of Java programming and how to use Java to write applications.
2. To train the learners to implement and use inheritance and polymorphism, including interfaces and abstract classes, Packages.
3. To make the students to design appropriate Exception Handling in Java methods.
4. To make the students to understand the concepts of Threads, Files and I/O Streams, Applets in java.

Course Outcomes:

1. It is expected that the students are competent enough to write, debug, and document well-structured Java applications.
2. It is expected that the learners implement Java classes from specifications.
3. It is expected that the learners effectively create and use objects from predefined class libraries.
4. It is expected that the students would apply decision and iteration control structures to implement algorithms.
5. It is expected that the students have proficient knowledge in handling of interfaces, inheritance, and polymorphism as programming techniques, Exceptions and Threads
6. It is expected that the students would work with Files & I/O Streams and Applets.

UNIT – I

Introduction to OOP

Introduction, Need of Object Oriented Programming, Principles of Object Oriented Languages, C++ vs Java, Applications of OOP, History of JAVA, Java Virtual Machine, Java Features, Program structures, Installation of JDK1.6 Variables , Primitive Data types, Identifiers- Naming Conventions, Keywords, Literals

UNIT - II

Programming Constructs

Operators- Binary, Unary and ternary, Expressions, Precedence rules and Associative, Primitive Type Conversion and Casting, Flow of control- 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 and Enumeration:

Interface-Extending interface, Interface vs. Abstract classes, Packages-Creating packages, using Packages, Access protection, java.lang package

UNIT – IV

Exceptions & Assertions –

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

Multi-Threading:

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

UNIT – V

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

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

UNIT – VI

Event Handling

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

Abstract Window Toolkit

Why AWT?, java.awt package, Components and Containers, Button, Label, Checkbox, Radio buttons, List boxes, Choice boxes, Text field and Text area, container classes, Layouts, Menu, Scroll bar

TEXT BOOKS:

1. The Complete Reference Java, 9ed, Herbert Schildt, TMH
2. Programming in JAVA, Sachin Malhotra, Saurabh choudhary, Oxford.

REFERENCE BOOKS:

1. JAVA Programming, K.Raj kumar. Pearson
2. Object oriented programming with JAVA, Essentials and Applications, Raj KumarBuyya, Selvi, Chu TMH
3. Introduction to Java Programming, 7/e, Y Daniel Liang, Pearson.
4. Core Java Volume 1.Fundamentals, 8ed, Cay S.Horstmann, Gray Cornell, Pearson.
5. Advanced Programming in Java2: Updated to J2SE6 with Swing, Servlet and RMI, K.Somasundaram.
6. A Java Programming Book by N.B.Venkateswarlu

WEB LINKS:

1. https://www.tutorialspoint.com/java/java_object_classes.htm
2. http://beginnersbook.com/2015/07/java-swing-tutorial/
3. http://www.realapplets.com/tutorial/
4. https://www.youtube.com/watch?v=aUlwgdakBug
5. http://beginnersbook.com/2013/04/java-exception-handling/
   and words in a text file.

SYLLABUS

II Year - II Semester

L	T	P	C
4	1	0	3

ADVANCED DATA STRUCTURES
(Computer Science & Engineering)

Learning objectives:

1. Comprehensive understanding of dictionaries, hashing mechanism which supports faster data retrieval and skip lists.
2. Illustration of Balanced trees and their operations.
3. Comprehension of heaps, queues and their operations Priority Queues.
4. Detailed knowledge of nonlinear data structures and various algorithms using them Graph algorithms.
5. Illustration of pattern matching and tries which share some properties of table look up.
6. Understanding of external sorting methods and various issues related to the design of file structures.

Course Outcomes

1. Understand implementation of symbol table using hashing techniques
2. Develop and analyze algorithms for Height balanced trees such as AVL trees, red-black trees, B-trees and Splay trees.
3. Understand implementation of priority queues using Binary heap and Binomial Queue and their applications
4. Identify suitable data structures and develop algorithms for representing graphs, Minimum-Cost Spanning Trees and Shortest Path problems
5. Develop algorithms for text processing applications
6. Understand implementation of external sorting methods and file structure concepts

UNIT-I: SORTING

External Sorting, Introduction, K-way Merging - Buffer Handling for parallel Operation- Run Generation- Optimal Merging of Runs.

UNIT-II: HASHING

Introduction-Static Hashing- Hash Table- Hash Functions- Secure Hash Function- Overflow Handling- Theoretical Evaluation of Overflow Techniques, Dynamic Hashing- Motivation for Dynamic Hashing -Dynamic Hashing Using Directories- Directory less Dynamic, Hashing.

UNIT-III: PRIORITY QUEUES (HEAPS)

Model, Simple Implementation, Binary Heap-Structure Property-Heap-Order Property-Basic Heap Operations- Other Heap Operation, Applications of Priority Queues- The Selection Problem Event Simulation Problem, Binomial Queues- Binomial Queue Structure – Binomial Queue Operation- Implementation of Binomial Queues

UNIT-IV: EFFICIENT BINARY SEARCH TREES

Optimal Binary Search Trees, AVL Trees, Red-Black Trees, Definition- Representation of a Red- Black Tree- Searching a Red-Black Tree- Inserting into a Red Black Tree- Deletion from a Red-Black Tree- Joining Red-Black Trees, Splitting a Red-Black tree.

UNIT-V: MULTIWAY SEARCH TREES

M-Way Search Trees, Definition and Properties- Searching an M-Way Search Tree, B-Trees, Definition and Properties- Number of Elements in a B-tree- Insertion into B-Tree- Deletion from a B-Tree- B+-Tree Definition- Searching a B+-Tree- Insertion into B+-tree- Deletion from a B+-Tree.

UNIT-VI: DIGITAL SEARCH STRUCTURES

Digital Search Trees, Definition- Search, Insert and Delete- Binary tries and Patricia, Binary Tries, Compressed Binary Tries- Patricia, Multiway Tries- Definitions- Searching a Trie- Sampling Strategies- Insertion into a Trie- Deletion from a Trie- Keys with Different Length- Height of a Trie- Space Required.

TEXT BOOKS:

1. Fundamentals of DATA STRUCTURES in C: 2nd ed. Horowitz , Sahani, Anderson-freed, Universities Press
2. Data structures and Algorithm Analysis in C, 2nd ed. Mark Allen Weiss, Pearson

REFERENCE BOOKS:

1. Data Structures, a Pseudocode Approach, Richard F Gilberg, Behrouz A Forouzan, Cengage.
2. File Structures :An Object oriented approach with C++, 3rd ed, Michel J Folk, Greg Riccardi, Bill Zoellick
3. C and Data Structures: A Snap Shot oriented Treatise with Live examples from Science and Engineering, NB Venkateswarlu & EV Prasad, S Chand, 2010.

URLs

1. 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. http://lcm.csa.iisc.ernet.in/dsa/dsa.html
6. http://utubersity.com/?page_id=878
7. http://freevideolectures.com/Course/2519/C-Programming-and-Data-Structures
8. http://freevideolectures.com/Course/2279/Data-Structures-And-Algorithms

SYLLABUS

II Year - II Semester

L	T	P	C
4	1	0	3

COMPUTER ORGANIZATION
(Common for CSE & IT)

Learning objectives

Comprehensive knowledge of computer system including the analysis and design of components of the system-

1. Gives a view of computer system from user's perspective, representation of data
2. Understanding RTL, Micro operations, ALU, organization of stored program computer, types of instructions and design of basic components of the system
3. Illustration of data paths and control flow for sequencing in CPUs, Microprogramming of control unit of CPU
4. Illustration of algorithms for basic arithmetic operations using binary and decimal representation
5. Description of different parameters of a memory system, organization and mapping of various types of memories
6. Describes the means of interaction devices with CPU, their characteristics, modes and introduction to multiprocessors.

Course Outcomes

1. Students can understand the architecture of modern computer.
2. Students should be able to demonstrate the programming proficiency using the various addressing modes and data transfer instructions of the target computer.
3. Understand different instruction types.
4. Understand how a computer performs arithmetic operation of positive and negative numbers.
5. Students can understand how cache mapping occurs in a computer and can solve various problems related to this.
6. Students should be able to know the circuitry to the processor I/O ports in order to interface the processor to the external devices.

UNIT - I

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-BCD-8421, 2421, excess-3, gray and excess-3 gray, error detection codes.

UNIT - II

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, Instruction cycle, Memory – Reference Instructions. Input – Output and Interrupt.

UNIT - III

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.

UNIT - IV

Computer Arithmetic: Addition and subtraction, multiplication algorithms, division algorithms, floating – point arithmetic operations.

UNIT - V

The Memory System: Memory hierarchy, main memory, auxiliary memory, associative memory, cache memory, virtual memory.

Pipelining Arithmetic and Instruction Pipeline, Basics of vector processing and Array Processors.

UNIT-VI

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.Morris Mano, 3/e, Pearson/PHI
2. Computer Organization, Carl Hamacher, Zvonks Vranesic, SafeaZaky, 5/e, McGraw Hill.

References

1. Computer Organization and Architecture – William Stallings, 6/e, Pearson/PHI
2. Structured Computer Organization – Andrew S. Tanenbaum, 4/e, PHI/Pearson
3. Fundamentals or Computer Organization and Design, - Sivaraama Dandamudi Springer Int. Edition.
4. Computer Architecture a quantitative approach, John L. Hennessy and David A. Patterson, 4/e, Elsevier

URLs

1. **http://nptel.iitm.ac.in/video.php?subjectId=106106092**
2. **https://www.tutorialspoint.com/videos/computer_organization/index.htm**
3. **https://www.reference.com/technology/computer-organization-36c3a064b20f9b33**
4. **https://www.youtube.com/watch?v=CDO28Esqmcg&list=PLhwVAYxlh5dvB1MkZrcRZy6x_a2yORNAu**

SYLLABUS

II Year - II Semester

L	T	P	C
4	1	0	3

FORMAL LANGUAGES AND AUTOMATA THEORY

(Computer Science & Engineering)

Learning objectives

1. To provide students with an understanding of basic concepts in the theory of computation.
2. Identify different formal language classes and their relationships
3. Design grammars and recognizers for different formal languages
4. Prove or disprove theorems in automata theory using its properties
5. Determine the decidability and intractability of computational problems.

Course Outcomes

1. Analysis of Finite state machine, its representation and automata
2. Description of finite automata, variants in it and their equivalence, Minimization, optimization of finite automata
3. Regular expressions and equivalence of finite automata and regular expressions
4. Illustration about grammars, classification and simplification of grammars
5. Description of pushdown automata and its representation
6. Delineation of Turing machines, decidable & undecidable problems

UNIT – I: Finite Automata

Why Study Automata Theory? The Central Concepts of Automata Theory, Automation, Finite Automation, Transition Systems, Acceptance of a String by a Finite Automation, DFA, Design of DFAs, NFA, Design of NFA, Equivalence of DFA and NFA, Conversion of NFA into DFA, Finite Automata with E-Transition, Minimization of Finite Automata, Mealy and Moore Machines, Applications and Limitation of Finite Automata.

UNIT – II: Regular Expressions

Regular Expressions, Regular Sets, Identity Rules, Equivalence of two Regular Expressions, Manipulations of Regular Expressions, Finite Automata, and Regular Expressions, Inter Conversion, Equivalence between Finite Automata and Regular Expressions, Pumping Lemma, Closers Properties, Applications of Regular Expressions, Finite Automata and Regular Grammars, Regular Expressions and Regular Grammars.

UNIT – III: Context Free Grammars

Formal Languages, Grammars, Classification of Grammars, Chomsky Hierarchy Theorem, Context Free Grammar, Leftmost and Rightmost Derivations, Parse Trees, Ambiguous Grammars, Simplification of Context Free Grammars-Elimination of Useless Symbols, E-Productions and Unit Productions, Normal Forms for Context Free Grammars-Chomsky Normal Form and Greibach Normal Form, Pumping Lemma, Closure Properties, Applications of Context Free Grammars.

UNIT – IV: Pushdown Automata

Pushdown Automata, Definition, Model, Graphical Notation, Instantaneous Description Language Acceptance of pushdown Automata, Design of Pushdown Automata, Deterministic and Non – Deterministic Pushdown Automata, Equivalence of Pushdown Automata and Context Free Grammars Conversion, Two Stack Pushdown Automata, Application of Pushdown Automata.

UNIT – V: Turing Machine

Turing Machine, Definition, Model, Representation of Turing Machines-Instantaneous Descriptions, Transition Tables and Transition Diagrams, Language of a Turing Machine, Design of Turing Machines, Techniques for Turing Machine Construction, Types of Turing Machines, Church's Thesis, Universal Turing Machine, Restricted Turing Machine.

UNIT – VI: Computability

Decidable and Un-decidable Problems, Halting Problem of Turing Machines, Post's Correspondence Problem, Modified Post's Correspondence Problem, Classes of P and NP, NP-Hard and NP-Complete Problems.

TEXT BOOKS:

1. Introduction to Automata Theory, Languages and Computation, J.E.Hopcroft, R.Motwani and J.D.Ullman, 3rd Edition, Pearson, 2008.
2. Theory of Computer Science-Automata, Languages and Computation, K.L.P.Mishra and N.Chandrasekharan, 3rd Edition, PHI, 2007.

REFERENCE BOOKS:

1. Formal Language and Automata Theory, K.V.N.Sunitha and N.Kalyani, Pearson, 2015.
2. Introduction to Automata Theory, Formal Languages and Computation, Shyamalendu Kandar, Pearson, 2013.
3. Theory of Computation, V.Kulkarni, Oxford University Press, 2013.
4. Theory of Automata, Languages and Computation, Rajendra Kumar, McGraw Hill, 2014.

URLs

1. http://nptel.ac.in/courses/106106049/
2. https://www.iitg.ernet.in/dgoswami/Flat-Notes.pdf
3. http://www.ics.uci.edu/~goodrich/teach/cs162/notes/
4. https://www.tutorialspoint.com/automata_theory/automata_theory_tutorial.pdf
5. http://freevideolectures.com/Course/3379/Formal-Languages-and-Automata-Theory

PRAGATI ENGINEERING COLLEGE
(Autonomous)
1-378, ADB Road, Surampalem, E.G.District, A.P. - 533 437
(Approved by AICTE & Permanently Affiliated to JNTUK, Kakinada & Accredited by NAAC with 'A' Grade)
(Recognised by UGC Under Sections 2 (f) and 12 (b) of UGC act, 1956)
Ph : 08852 - 252233, 252234, 252235, Fax : 252232, Website : www.pragati.ac.in
(Sponsored by Gayatri Educational Society)
D.No. 2-24-4/2, Ground Floor, Janmabhoomi Park Road, Srinagar, Kakinada - 3, Ph : 0884 - 2355900, Fax : 2363900

Department of Computer Science and Engineering

SYLLABUS

II Year - II Semester

L	T	P	C
4	1	0	3

COMPUTER GRAPHICS
(Computer Science & Engineering)

Learning objectives

1. To make students understand about fundamentals of Graphics
2. To make the student present the content graphically.

Course Outcomes

1. The algorithms for drawing line, circle and ellipse which improves the programming capabilities in graphics
2. Apply algorithms for filling of closed regions.
3. 2-D transformation techniques, 2-D viewing and apply algorithms for clipping of 2D graphic primitives
4. Get knowledge on 3-D objects
5. Get knowledge on 3-D transformations and viewing
6. Motion oriented graphics will give the idea about implementing different animation sequences.

UNIT - I

Introduction: Application of Computer Graphics, raster scan systems, random scan systems, raster scan display processors.

Output primitives : Points and lines, line drawing algorithms (Bresenham's and DDA Line derivations and algorithms), mid-point circle and ellipse algorithms.

UNIT – II

Filled area primitives: Scan line polygon fill algorithm, boundary-fill and flood-fill algorithms. Inside and outside tests.

UNIT – III

2-D geometrical transforms: Translation, scaling, rotation, reflection and shear transformations, matrix representations and homogeneous coordinates, composite transforms, transformations between coordinate systems.

2-D viewing: The viewing pipeline, viewing coordinate reference frame, window to view-port coordinate transformation, viewing functions, Cohen-Sutherland and Cyrus-beck line clipping algorithms, Sutherland –Hodgeman polygon clipping algorithm.

UNIT – IV

3-D object representation: Polygon surfaces, quadric surfaces, spline representation, Hermite curve, Bezier curve and B-Spline curves, Bezier and B-Spline surfaces.

UNIT - V

3-D Geometric transformations: Translation, rotation, scaling, reflection and shear transformations, composite transformations. 3D Viewing pipeline, clipping, projections (Parallel and Perspective).

UNIT – VI

Graphics Programming

Color Models – RGB, YIQ, CMY, HSV – Animations – General Computer Animation, Raster, Key frame - Graphics programming using OPENGL – Basic graphics primitives – Drawing three dimensional objects - Drawing three dimensional scenes.

Text Books

1. Computer Graphics *C* version, Donald Hearn, M.Pauline Baker, Pearson
2. Computer Graphics with Virtual Reality Systems, Rajesh K Maurya, Wiley
3. Introduction to Computer Graphics, Using Java 2D and 3D, Frank Klawonn, Springer
4. Computer Graphics, Steven Harrington, TMH
5. Computer Graphics, Amarendra N Sinha, Arun Udai, TMH

Reference Books

1. Computer Graphics Principles & practice, 2/e, Foley, VanDam, Feiner, Hughes, Pearson
2. Computer Graphics, Peter, Shirley, CENGAGE
3. Principles of Interactive Computer Graphics, Neuman , Sproul, TMH.
4. The Computer Graphics manual, Vol 2, David, Soloman, Springer
5. Procedural elements for Computer Graphics, David F Rogers, 2/e, TMH

URLs

1. https://www.ssmengg.edu.in/weos/weos/upload/EStudyMaterial/Cse/6th%20sem/computer%20graphics.pdf
2. https://www.tutorialspoint.com/**computer_graphics**/

SYLLABUS

II Year - II Semester

L	T	P	C
0	0	3	2

ADVANCED DATA STRUCTURES LAB
(Computer Science & Engineering)

Learning objectives

1. To make the student learn about object oriented way of solving problems.
2. To make the student understand and implement operations for all data structures.
3. To make learners how to develop algorithms for text processing applications.

Course Outcomes

1. Implement functions of Dictionary using Hashing.
2. Develop programs for implementing balanced trees and their operations.
3. Implement graph and graph traversal algorithms.
4. Develop programs for text processing applications.

List of Programs

1. Implementation of Merge Sort
2. Implementation of Quick Sort
3. To implement operations on graphs
    a. Vertex insertion
    b. Vertex deletion
    c. Finding vertex
    d. Edge addition and deletion
4. To implement Depth First Search for a graph non recursively.
5. To implement Breadth First Search for a graph non recursively.
6. To implement functions of Dictionary using Hashing (division method, Multiplication method, Universal hashing)
7. To perform various operations i.e., insertions and deletions on AVL trees
8. To perform various operations i.e., insertions and deletions on 2-3 trees.

9. To implement operations on binary heap.
10. Implementation of Data Searching using divide and conquer technique
11. To implement pattern matching using Boyer-Moore algorithm.
12. To implement Knuth-Morris-Pratt algorithm for pattern matching.

PRAGATI ENGINEERING COLLEGE
(Autonomous)
1-378, ADB Road, Surampalem, E.G.District, A.P. - 533 437
(Approved by AICTE & Permanently Affiliated to JNTUK, Kakinada & Accredited by NAAC with 'A' Grade)
(Recognised by UGC Under Sections 2 (f) and 12 (b) of UGC act, 1956)
Ph : 08852 - 252233, 252234, 252235, Fax : 252232, Website : www.pragati.ac.in
(Sponsored by Gayatri Educational Society)
D.No. 2-24-4/2, Ground Floor, Janmabhoomi Park Road, Srinagar, Kakinada - 3, Ph : 0884 - 2355900, Fax : 2363900

Department of Computer Science and Engineering

SYLLABUS

II Year - II Semester

L	T	P	C
0	0	3	2

JAVA PROGRAMMING LAB
(Common to CSE & IT)

Learning Objectives

1. To make the students understand how to use Java to write applications.
2. To make the students implement object-oriented concepts, including classes, objects, methods, properties, abstraction, polymorphism, inheritance, encapsulation, and more.
3. To make the learners implement and use inheritance and polymorphism, including interfaces and abstract classes, Packages.
4. To make the learners proficient in designing appropriate Exception Handling using Java methods.
5. To make students understand Threads, Files and I/O Streams, Applets in java

Course Outcomes

1. It is expected that the learners write, debug, and document well-structured Java applications effectively.
2. It is expected that students implement Java classes from specifications effectively.
3. It is expected that the learners effectively create and use objects from predefined class libraries.
4. It is expected that the students apply decision and iteration control structures to implement algorithms effectively.
5. It is expected that the learners are proficient in Handling of interfaces, inheritance, and polymorphism as programming techniques, Exceptions and Threads
6. It is expected that the students write and work with Files & I/O Streams and Applets.

Week-1:

a) Write a JAVA program to display default value of all primitive data types of JAVA
b) Write a JAVA program to display the Fibonacci sequence

Week-2:

a) Write a JAVA program to check whether given string is palindrome or not.
b) Write a JAVA program to sort given list of numbers.

Week-3:

a)Write a JAVA program give example for command line arguments.
b)Write a JAVA program to sort an array of strings

Week-4

a) Write a JAVA program for the following
1. Example for call by value.
2. Example for call by reference.

b) Write a JAVA program to give the example for 'this' operator. And also use the 'this' keyword as return statement

Week-5

a) Write a JAVA program to demonstrate static variables, methods and blocks,

b) Write a JAVA program to give the example for 'super' keyword.
c) Write a JAVA program demonstrating the difference between method overloading and method overriding.
d) Write a JAVA program demonstrating the difference between method overloading and constructor overloading.

Week-6

a) Write a JAVA program that illustrates multi-level inheritance
b) Write a JAVA program illustrating multiple inheritance using interfaces.
c) Write a JAVA program to give a simple example for abstract class.

Week-7:

a) Write a JAVA program to create a package named pl, and implements this package in ex 1 class.
b) Write a JAVA program to create a package named my pack and import it in circle class.

Week-8:

a) Write a JAVA program to illustrate sub class exception precedence over base class.
b) Write a JAVA program for example of try and catch block. In this check whether the given array size is negative or not.

c) Write a JAVA program for creation of user defined exception

Week-9:

a) Write a JAVA program to illustrate creation of threads using Runnable interface. (Start method starts each of the newly created thread. Inside the run method there is sleep() for suspend the thread for 500milliseconds).

b) Write a JAVA program to create a class My thread 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.

Week-10:

a) Write a JAVA program that describes the life cycle of an applet.
b) Write a JAVA program to create a dialog box and menu.

Week-11:

a) Write a JAVA program to create a border layout control.
b) Write a JAVA program to create a simple calculator.

Week-12:

a) Write a JAVA program that displays that x and y position of the cursor movement using Mouse.

b) Write a JAVA program that displays number of characters, lines and words in a text file.