Program Structure and Syllabus of B. Tech II Year (I & II Semesters)

Computer Science and Engineering

R20 Regulations

Venkatapur (V), Ghatkesar (M), Medchal-Malkajgiri (Dt.),
Hyderabad, Telangana, INDIA
info@anurag.edu.in; http://anurag.edu.in

B.TECH II YEAR I SEMESTER [5 T + 4 P + 1 M]

S. No	Course Code	Category	Course	Hours per week			Credits
				L	T	P	
1	A53024	ESC	Digital Logic Design	3	0	0	3.0
2	A53025	PCC	Data Structures	3	0	0	3.0
3	A53027	BSC	Discrete Mathematics	3	0	0	3.0
4	A53028	PCC	Formal Languages and Automata Theory	2	1	0	3.0
5	A53026	PCC	Python Programming	2	0	0	2.0
6	A53211	PCC LAB	Python Programming Lab	0	0	3	1.5
7	A53212	PCC LAB	Data Structures Lab	0	0	3	1.5
	A53213	PCC LAB	Linux Programming Lab	0	1	2	2.0
8	A53214	ESC LAB	Design Thinking Lab	0	0	2	1.0
9	A53007	MC	Gender Sensitization	2	0	0	0
			TOTAL	15	2	8	20

B. TECH II YEAR II SEMESTER [5 T + 3 P + 1 M]

S.No	Course Code	Category	Course	Hours per week			Credits
				L	T	P	
1	A54023	PCC	Computer Organization and Architecture	3	0	0	3.0
2	A54024	BSC	Probability and Statistics	3	0	0	3.0
3	A54025	PCC	Java Programming	2	1	0	3.0
4	A54026	PCC	Database Management Systems	3	0	0	3.0
5	A54027	PCC	Design and Analysis of Algorithms	3	1	0	4.0
6	A54214	PCC LAB	Database Management Systems Lab	0	0	3	1.5
7	A54215	PCC LAB	Java Programming Lab	0	0	3	1.5
8	A54216	HSS & MC LAB	Soft Skills for Success Lab	0	0	2	1.0
9	A54022	MC	Environmental Science	2	0	0	0
			TOTAL	16	2	8	20

Data Structures

B. Tech II Year I Semester					Dept. of Computer Science and Engineering			
Code	**Category**	**Hours / Week**			**Credits**	**Marks**		
	PCC	L	T	P	C	CIE	SEE	Total
		3	0	0	3	40	60	100

Pre requisites

Any Programming Language

Course Objectives

1. Understand various static and dynamic representations of data structures
2. Understand fundamental algorithmic problems of various nonlinear data structures.
3. To be familiar with Graph representations and traversals.
4. Know the basic concepts of Hashing.

Course Outcomes

1. Examine Static and Dynamic data structures in implementing Stack applications (L4)
2. Apply Tree traversal algorithms in solving real time applications (L3)
3. Analyze the concepts of Advanced Trees to generate search efficiently (L4)
4. Interpret the importance of Graphs in solving real time applications (L5)
5. Examine the concepts of hashing, collision and its resolution methods using hash function (L4)

UNIT I

Introduction: What is data structure, Types of data structures, Static and Dynamic representation of data structure and comparison. Stacks-Definition, Operations, Applications of stacks – Representation and evaluation of expressions using Infix, Prefix and Postfix, Algorithms for conversions and evaluations of expressions from infix to prefix and postfix using stack, Towers of Hanoi, Parenthesis checker.

UNIT II

Trees: Basic terminology, Types of trees: Binary Tree: terminology, Complete and Full Binary Tree, Extended Binary Trees, Threaded Binary Trees-Inorder Threading. Representation of Trees using Arrays and Linked lists (advantages and disadvantages). Tree Traversal and Representation of Algebraic expressions; Algorithms for Tree Traversals.

Heaps: Introduction, Types of Heaps – Min binary heap, Max binary heap.

UNIT III

Advanced concepts on Trees: Representation and Creation of Binary Search Trees (BST), Algorithm for inserting, deleting and searching in BST. Representation and advantages of AVL Trees, Algorithms on AVL Trees-Insertion, Rotation and Deletion. Definition and advantages of B-trees, B Tree of Order M, operations- Insertion and Searching, Introduction to Red-Black Trees and Splay Trees.

UNIT IV

Graphs: Basic terminology, Representation of Graphs: sequential representation (Adjacency, Path Matrix) Linked representation.
Graph Traversals-Breadth First Search, Depth First Search with algorithms. Definition and properties of Spanning Tree, Minimum Spanning Tree, Minimum Spanning Tree Algorithms, Dijkstra Algorithms.

UNIT V

Hashing: General Idea, Hash Functions, Collision Resolution- Separate Chaining, Open Addressing-Linear probing, Quadratic Probing, Double Hashing, Rehashing, Extendible Hashing, Implementation of Dictionaries.

Text Book

1. Seymour Lipschutz, Schaum's Outlines, Data Structures, Special Second Edition, Tata McGraw-Hill, 2014.

Reference Books

1. Richard F.Gillberg & Behrouz A. Forouzan, Data Structures, A Pseudo code Approach with C, Second Edition, Cengage Learning, India Edition, 2005.
2. Aaron M. Tanenbaum, Yedidyah Langsam and Moshe J. Augenstein, Data Structures Using C and C++, PHI Learning Private Limited, Delhi India, 2001.
3. Horowitz and Sahani, Fundamentals of Data Structures, Galgotia Publications Pvt Ltd. Delhi India, 2015.
4. A.K. Sharma, Data Structure Using C, Pearson Education India, 2011

Digital Logic Design

B. Tech II Year I Semester					Dept. of Computer Science and Engineering			
Code	**Category**	**Hours / Week**			**Credits**	**Marks**		
	ESC	L	T	P	C	CIE	SEE	Total
		3	0	0	3	40	60	100

Pre requisites

None

Course Objectives

1. Understand various number systems addition and subtractions in binary system, error detection and correction codes
2. Minimize boolean functions using boolean laws & k-maps and realize by using logic gates
3. Design various combinational circuits with practical applications
4. Understand the basic sequential circuits : Latches, Flip-Flops and their usage
5. Design synchronous and asynchronous counters

Course Outcomes

1. Understand various number systems, floating point representations, complements, error detecting and correcting codes (L2)
2. Apply boolean algebraic principles and k-maps for simplification of boolean functions (L3)
3. Design combinational circuits (L3)
4. Analyze various types of flip flops (L4)
5. Design sequential circuits (L3)

UNIT I

Number Systems: Binary, Octal, Hex Decimal, and Conversions; 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: 8421, 2421, Ex-3, Gray and Self Complementary codes; Error Detecting codes: even & odd parity, hamming codes; Error correcting codes: hamming codes, block parity codes; Floating point representation

UNIT II

Boolean Algebra and Digital Logic Gates, Basic Boolean laws and properties; Boolean functions, truth tables; Standard forms (SOP, POS) and Canonical forms, Conversion

between Canonical and Standard forms ; Gate minimization using three and four variable K-Maps with and without don't cares, Logic Circuit Design using Universal Gates

UNIT III

Introduction to combinational circuits and applications**,** Design Procedure, Combinational circuit for Half Adder, Full Adder, Half Subtractor and Full Subtractor, Binary Adder, Binary Adder-Subtractor, Decimal Adder, Code Converters, Decoders, Encoders, Multiplexers, Demultiplexers

UNIT IV

Introduction to Sequential Circuits and its applications, Lathes, Flip flops, Storage Elements, Flip-flops: S-R Flip flop, D Flip Flop, J-K Flip Flop, T Flip flop, master slave J-K flip flop, Analysis of Clocked Sequential Circuits, Flip Flop Conversions

UNIT V

Registers and Counters: Introduction, Registers, Shift Registers, Ripple Counters: Up counter, Up-Down counter, Decade counter, Synchronous Counters: Up Counter, Up-Down counter, Decade Counter, Other Counters: Ring Counter, Johnson Counter

Text Books:

1. M. Morris Mano and Michael D. Ciletti, Digital Design, 5th Edition, Pearson Education, 2012
2. Anand Kumar, Switching Theory and Logic Design, 3rd edition, PHI, 2016

Reference Books

1. Roth, Fundamentals of Logic Design, 5th Edition, Thomson, 2004.
2. John F. Wakerly, Digital Design, Principles and Practices, 4th Edition, Pearson / Prentice Hall, 2005.
3. Malvino & Leach, Digital Principles and Applications, Seventh Edition, Tata McGraw-Hill Education, 2010.
4. A.K. Maini, Digital Electronics, Principles and Integrated Circuits, 1st Edition, Wiley India Publications, 2007.

Discrete Mathematics

B. Tech II Year I Semester					Dept. of Computer Science and Engineering			
Code	**Category**	**Hours / Week**			**Credits**	**Marks**		
	BSC	L	T	P	C	CIE	SEE	Total
		3	0	0	3	40	60	100

Pre requisites

Mathematics I and Mathematics II

Course Objectives

1. Interpret the Sets, syntax and semantics of propositional and predicate logic.
2. Solve applications involving Permutations and Combinations.
3. Formulate Recurrence relations to solve problems involving an unknown sequence.
4. Explain the concepts of Relations and Graphs.
5. Illustrate the Algebraic Systems

Course Outcomes

1. Analyze Statement Logic and Predicate Logic.(L4)
2. Apply the principles of Permutations and Combinations with repetition & without repetitions(L3)
3. Solve Recurrence Relations by using generating functions(L3)
4. Apply the knowledge of Relations and Graph Theory in the field of Computer Science.(L3)
5. Analyze the Algebraic Systems with their properties(L4)

UNIT I

Foundations: Basics, Sets and Operations of Sets, Fundamentals of Logic, Logical Inferences, First order logic and other methods of Proof, Rules of Inference for Quantified Propositions. **(Problems Only and Theorems without Proofs)**

UNIT II

Elementary Combinatorics: Basics of Counting, Combinations and Permutations, Enumerating Combinations and Permutations with & without repetitions, constrained repetitions, and Principle of Inclusion and Exclusion. **(Problems Only and Theorems without Proofs)**

UNIT III

Recurrence Relations: Generating Functions, Calculating coefficient of Generating Function, Solving Recurrence relations by substitution method and Generating Functions, The Method of Characteristic Roots, Solutions to inhomogeneous recurrence relations. **(Problems Only and Theorems without Proofs)**

UNIT IV

Relations and Digraphs: Relations and Directed Graphs, Special Properties of Binary Relations, Equivalence Relations, Ordering Relations, Lattices, Operations on Relations, Paths and Closures, Directed Graphs and adjacency matrices. **(Problems Only and Theorems without Proofs)**

Graphs: Basic Concepts, Isomorphism's and Sub-graphs, Planar Graphs, Euler's Formula, Multi-graphs and Euler Circuits, Hamiltonian Graphs. **(Problems Only and Theorems without Proofs)**

UNIT V

Algebraic structures: Algebraic systems, examples and general properties, semi groups and monoids, groups, sub groups, homomorphism, isomorphism, rings. **(Problems Only and Theorems without Proofs)**

Text Books

1. Joe L. Mott, Abraham Kandel, Theodare P. Baker, "Discrete Mathematics for Computer Scientists and Mathematicians", Second Edition, PHI, 2019.
2. J. P.Tremblay and P. Manohar, "Discrete Mathematical Structures with Applications to Computer Science", Tata McGraw Hill, 2007

Reference Books

1. K. H. Rosen, "Discrete Mathematics and its Applications with Combinatorics and Graph Theory", 7th Edition, Tata McGraw Hill.
2. S. K. Chakraborthy and B.K. Sarkar," Discrete Mathematics ", Oxford, 2011.
3. C. L. Liu and D. P. Mohapatra,"Elements of Discrete Mathematics-A Computer Oriented Approach", 3rd Edition, Tata McGraw Hill.

.

Formal Languages and Automata Theory

B. Tech II Year I Semester					Dept. of Computer Science and Engineering			
Code	**Category**	**Hours / Week**			**Credits**	**Marks**		
	PCC	L	T	P	C	CIE	SEE	Total
		2	1	0	3	40	60	100

Pre requisites

Basics of any Programming Language

Course Objectives

1. Summarize the concepts of Formal Languages and different kinds of Finite Automata
2. Interpret capabilities of Context Free Grammar.
3. Identify the significance of Push Down Automata.
4. Categorize various grammars of Regular Language
5. Outline the importance of Turing Machines.

Course Outcomes

1. Design of regular expressions for language constructs and conversions of NFA to DFA
2. Demonstrate the derivations and properties of context free grammars.
3. Analyze the applications of pushdown automata.
4. Construct DFA for Right Linear Grammar and Left Linear Grammar.
5. Appreciate the role of the Turing machine as computational and universal machine.

UNIT I

Fundamental concepts: Strings, Alphabets, Language operations, Regular Expressions, Regular Languages: Finite automata, Types of finite automata (FA)-Non deterministic Finite Automata (NFA), Deterministic Finite Automata(DFA), NFA with ε-Moves, regular expression representation; Regular expressions to NFA; NFA with ε-Moves to NFA without ε-Moves; NFA to DFA Conversions; Minimization of DFA (Proofs Not Required)

UNIT II

DFA with more than two outputs: Moore and Melay machines, Pumping Lemma for Regular Sets: Closure properties of Regular Sets (Proofs Not Required): Context Free

Grammars (CFG), Right most, Left most –derivations, Parse Trees; Operator Grammar: Unit productions; Chomsky normal forms; (Proofs Not Required)

UNIT III

Left recursion and Elimination of left recursion in CFG: Elimination of useless symbols and unit productions; Greibach Normal Form, Push Down automata (PDA): Types of PDA: Design of a PDA for a given CFG. (Proofs Not Required)

UNIT IV

Regular Grammars (RG), Design of DFA for a given RG: Right linear and left linear Grammars and conversions: Definition of Context Sensitive Grammar (CSG) and Linear bounded automata (LBA) (Proofs Not Required).

UNIT V

Definition of unrestricted Grammar and Turing Machine (TM): Chomsky hierarchy on Languages, Grammars and recognizers; Design of TM as recognizer; Types of TM: Computational problems of TM with multiple tracks; Decidability Problem; Churches hypothesis (Proofs Not Required)

Text Books

1. John E.Hopcroft, Rajeev Motwani, Jeffrey D.Ullman, Introduction to Automata Theory, Languages and Computation, Third Edition, Pearson, 2013.
2. VivekKulakarni, Theory of Computation, Oxford University press 2013, Fifth Edition, 2018

Reference Books

1. Daniel I.A.Cohen, Introduction to Computer Theory, Second Edition, John Wiley,1996.
2. John C Martin, Introduction to languages and the theory of Computation, Third Edition, TATA McGraw Hill, 2014.

Python Programming

B. Tech II Year I Semester					Dept. of Computer Science and Engineering			
Code	**Category**	**Hours / Week**			**Credits**	**Marks**		
	PCC	L	T	P	C	CIE	SEE	Total
		2	0	0	2	40	60	100

Pre requisites

None

Course Objectives

1. Understand the basics and function of Python Programming Language.
2. Understand the string operation and sequences used in Python Programming Languages.
3. Understand the data structures used in Python Programming Languages.
4. Know the classes and objects in Python Programming Language.
5. Use the reusability concepts in Python Programming Language.

Course Outcomes

t the end of this Data Structures course, students will be able to:

1. Apply control structures, functions and packages in Problem Solving. (L3)
2. Analyze various String handling functions and data structures(L4)
3. Model the object-oriented problems with classes and objects (L4)
4. Solve the problems by using Inheritance and polymorphism (L3)
5. Illustrate programs on Exception Handling and various packages(L3)

UNIT I

Introduction to Python:
Features of Python Language, Data Types, Operators, Expressions, Control Statement, Standard I/O Operations.
Functions and Modules:
Declaration and Definition Function Calling, More on Defining Functions, Recursive Functions, Modules, Packages in Python, Doc Strings.

UNIT II

Strings and Regular Expressions:
String Operations, Built-in String Methods and Functions, Comparing Strings, function in Regular Expression.
Sequence: List, Tuples, Dictionaries, Sets.

UNIT III

Introduction to Object Oriented Programming: Features of OOP,Merits and demerits of Object Oriented Programming Languages,Applications of OOP

Implementation of classes and objects in Python:

Classes and Objects, Class Method and Self Argument. The __init__Method, Class Variables and Object Variables, The __del__Method, Public and Private Data Members, Private Methods, Built-in Functions to Check, Get, Set and Delete Class Attributes, Garbage Collection (Destroying Objects).

UNIT IV

Implementation of Inheritance in Python:

Inheriting Classes in Python, Types of Inheritance, Abstract Classes and Interfaces, Meta class,

Implementation of Operator Overloading in Python:

Introduction, Implementing Operator Overloading, Overriding Methods

Exception Handling in Python:

Introduction, Exception hierarchy, Handling Exception, Multiple Except Blocks and Multiple Exceptions, Finally Block.

UNIT V

Python NumPy: NumPy ND array, Data Types, Functions of NumPy Array, NumPy Array Indexing, Mathematical Functions on Arrays in NumPy

Python Pandas: Pandas Features, Dataset in Pandas, Data Frames, Manipulating the Datasets, Describing a Dataset, group by Function, Filtering, Missing Values in Pandas, Concatenating Data Frames. Import data from csv file.

Introduction to Matplotlib :, Plot, Scatterplot, Introduction to Tkinter ,Date and Time Packages

Text Books

1.ReemaThareja,Python Programming using Problem Solving Approach, First Edition,Oxford Higher Eduction,2017

2.James Payne, Beginning Python using Python 2.6 and Python 3,1st Edition

Reference Books

1. Charles Dierach, Introduction to Computer Science using Python,2013

2. https://www.programiz.com/python-programming

3. https://www.javatpoint.com/python-tutorial

4.. https://www.geeksforgeeks.org/python-programming-language/

GENDER SENSITIZATION

B. Tech II Year II Semester					Dept. of Computer Science and Engineering			
Code	**Category**	**Hours / Week**			**Credits**	**Marks**		
	PCC	L	T	P	C	CIE	SEE	Total
		2	0	0	0	--	--	--

Course Objectives

1. To develop students sensibility with regard to issues of gender in contemporary India.
2. To provide a critical perspective on the socialization of men and women.
3. To introduce students to information about some key biological aspects of genders.
4. To expose the students to debates on the politics and economics of work.
5. To help students reflect critically on gender violence.
6. To expose students to more egalitarian interactions between men and women

Course Outcomes

1. Students will have developed a better understanding of important issues related to gender in contemporary India.
2. Student will be sensitized to basic dimensions of the biological, sociological, psychological and legal aspects of gender. This will be achieved through discussion of materials derived from research, facts, everyday life, literature and film.
3. Students will attain a finer grasp of how gender discrimination works in our society and how to counter it.
4. Students will acquire insight into the gendered division of labour and its relation to politics and economics.
5. Men and women students and professionals will be better equipped to work and live together as equals.
6. Students will develop a sense of appreciation of women in all walks of life.
7. Through providing accounts of studies and movements as well as the new laws that provide protection and relief to women, the textbook will empower students to understand and respond to gender violence.

UNIT I

UNDERSTANDING GENDER: Gender: Why Should We Study It? (Towards a World of Equals: Unit-1) Socialization: Making Women Making Men (Towards a World of Equals: Unit-2), Introduction. Preparing for Womanhood. Growing up Male. First lesions in Caste. Different Masculinities. Just Relationships: Being Together as Equals (Towards a

World of Equals: Unit-12)Mary Kom and Onler. Love and Acid just do not Mix. Love Letters. Others and Fathers. Further Reading: Rosa Parks-The Brave Heart.

UNIT II

GENDER AND BIOLOGY: Missing Women: Sex Selection and Its Consequences, (Towards a World of Equals: Unit-4) Declining Sex Ratio. Demographic, Consequences. Gender Spectrum: Beyond the Binary (Towards a World of Equals: Unit-10)Two or Many? Struggles with Discrimination.Additional Reading: Our Bodies, Our Health (Towards a World of Equals: Unit-13)

UNIT III

GENDER AND LABOUR: Housework: the Invisible Labour (Towards a World of Equals: Unit-3)"My Mother doesn't Work." "Share the Load."Women's Work: Its Politics and Economics (Towards a World of Equals; Unit-7)Fact and Fiction. Unrecognized and Unaccounted work. Further Reading: Wages and Conditions of Work.

UNIT IV

ISSUES OF VIOLENCE: Sexual Harassment: Say No! (Towards a World of Equals: Unit-6) Sexual Harassment not Eve-Teasing- Coping with Everyday Harassment-Further Reading: "Chupulu". Domestic Violence: Speaking Out (Towards a World of Equals: Unit-8)Is Home a Safe Place? –When Women Unite (Film). Rebuilding Lives. Further Reading: New Forums for Justice. Thinking about Sexual Violence (Towards a World of Equals: Unit-11)Blaming the Victim-"I Fought for my Life…." – Further Reading: The Caste Face of Violence.

UNIT V

GENDER STUDIES: Knowledge: Through the Lens of Gender (Towards a World of Equals: Unit-5), Point of View. Gender and the Structure of Knowledge. Further Reading: Unaknowledged.Women Artists of Telangana.Whose History? Questions for Historians and Others (Towards a World of EqualsReclaiming a Past. Writing other Histories. Further Reading: Missing Pages from Modern Telangana History.Essential Reading: All the Units in the Textbook, "Towards a World of Eqals: A Bilingual Textbook on Gender" written by A.Suneetha, Uma Bhrugubanda, Duggirala Vasanta, Rama Melkote, Vasudha Nagarj, Asma Rasheed, Gogu Shyamala, Deepa Sreenivas and Susie Tharu.

Note: Since it is Interdisciplinary Course, Resource Persons can be drawn from the fields of English Literature or Sociology or Political Science or any other qualified faculty who has expertise in this field.

Reference Books

1. Sen, Amartya, “More than One Million Women are Missing.” New York Review of Books 37.20 (20 December 1990). Print. ‘We Were Making Hisoty…’ Life Stories of Women in the Telangana People’s Struggle. New Delhi: Kali for Women, 1989
2. Tripti Lahiri. “By the Numbers: Where Indian Women Work.” Women’s Studies Journal (14 November 2012) Available online at:http://blogs.wsj.com/ India real time/2012/11/14/by-the-numbers-where-Indian-women-work/>
3. K.Satyanarayana and Susie Tharu (Ed.) Steel Nibs Are Sprouting: New Dalit Writing From South India, Dossier 2, Telugu and Kannada http://harpercollings.co.in/BookDetail.asp?Book Code=3732

Python Programming Lab

B. Tech II Year I Semester					Dept. of Computer Science and Engineering			
Code	**Category**	**Hours / Week**			**Credits**	**Marks**		
	PCC LAB	L	T	P	C	CIE	SEE	Total
		0	0	3	1.5	50	50	100

Pre requisites

Python Programming

Course Objectives

1. Understand the basics and function of Python Programming Language.
2. Understand the string operation and sequences used in Python Programming Language.
3. Know the Data Structures in Python Programming Language.
4. Use the reusability concepts in Python Programming Language.
5. Use Exception Handling mechanism in Python Programming Language.
6. Know the packages in Python Programming Language

Course Outcomes

At the end of this course, students will be able to:

1. Develop programs on data types, operators and expressions
2. Apply the data structures in real time scenarios
3. Write the programs on strings and functions
4. Implement programs on class and related issues.
5. Use of python exception handling and packages.

Week 1

1. Installation and Environment set up of Python & Programs on Data types

Week 2

2.Programs on Standard I/O, Operators and Expressions

Week 3

Programs on Functions

Week 4

Programs on lists and Tuples

Week 5

Programs on Dictionaries

Week 6

Programs on Strings and string operations

Week 7

Programs on Regular Expressions.

Week 8
Programs on Inheritance and Polymorphism
Week 9
Programs on Exception Handling
Week 10
Demonstration of Numpy Package
Week 11
Demonstration of Pandas Package
Week 12
Demonstration of matplotlib Package and Tkinter Package
Week 13
Demonstration of Date and Time Packages
Week 14 and 15
Review

Data Structures Lab

B. Tech II Year I Semester					Dept. of Computer Science and Engineering			
Code	**Category**	**Hours / Week**			**Credits**	**Marks**		
	PCC LAB	L	T	P	C	CIE	SEE	Total
		0	0	4	2	50	50	100

Pre requisites

Data structures course

Course Objectives

1. To design and analyze simple linear and non linear data structures.
2. To design and implement various data structure algorithms
3. To identify and apply the suitable data structure for the given real world problem

Course Outcomes

1. Develop the programs on stacks and its applications.
2. Demonstrate the implementation of various advanced trees.
3. Design and implementation of programs on BST and Graph Traversals.
4. Develop the programs on Hashing and Dictionaries

Week 1

1. Review of Stack and Queue Operations using arrays and Linked Lists

Week 2

2. Program to convert infix to postfix notation
3. Program to evaluate postfix notations

Week 3

4. Program to implement towers of Hanoi
5. Program to implement parenthesis checker

Week 4

6. Program to illustrate tree traversals

a) In order b) Preorder c) Post order

Week 5

7. Program to illustrate insertion, deletion and searching in Binary Search Tree

Week 6

8. Program to implement Heaps

a) Min Heap b) Max Heap

Week 7

9. Program to illustrate Insertion on AVL Trees
10. Program to illustrate deletion and Rotation on AVL Trees

Week 8

11. Program to implement B-Trees

a) Insertion b) Search c) Display

Week 9

12. Program to illustrate Graph traversals

a) Breadth First Search
b) Depth First Search

Week 10

13. Program to implement

a) Prim's algorithm b) Kruskal's algorithm

Week 11

14. Program to Implement Dijkstra algorithm

Week 12 & 13

15. Program to implement Hashing and collision resolution techniques

Week 14

16. Program to implement Dictionaries

Week 15

17. Review

Linux Programming Lab

B. Tech II Year I Semester					Dept. of Computer Science and Engineering			
Code	**Category**	**Hours / Week**			**Credits**	**Marks**		
	PCC LAB	L	T	P	C	CIE	SEE	Total
		0	1	2	2	40	60	100

Pre requisites

Basic Computer fundamentals

Course Objectives

1. To gain an understanding of important aspects related to the Linux Commands.
2. To understand directory commands.
3. To provide a comprehensive introduction to SHELL programming.
4. To understand file handling utilities
5. To develop ability to use system calls.

Course Outcomes

1. Apply the basic commands in Linux Operating System.
2. Create directories and Shell Script programs.
3. Analyze a given problem and apply requisite facets of Shell programming.
4. Demonstrate UNIX commands for file handling mechanisms.
5. Develop a C Program for UNIX Commands.

Week 1

Practice Vi Commands

Week 2

Open the file created in session 1
Add some text
Change some text
Delete some text
Save the Changes

Week 3

a) Create mytable (name of the table) using cat command for the following data. use tab to separate fields.

1425 Ravi 15.65
4320 Ramu 26.27

6830 Sita 36.15
1450 Raju 21.86

b) Use the cat command to display the file, mytable.

c) Use the vi command to correct any errors in the file, mytable.

Week 4

a) Use the sort command to sort the file mytable according to the first field. Call the sorted file
mytable (same name)

b) Print the file mytable

c) Use the cut and paste commands to swap fields 2 and 3 of mytable. Call it my table (same name)

d) Print the new file, mytable

e) Logout of the system.

Week 5

a) Use the appropriate command to determine your login shell

b) Use the /etc/passwd file to verify the result of “step a”.

c) Use the who command and redirect the result to a file called myfile1. Use the more command
to see the contents of myfile1.

d) 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.

Week 6

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.

Week 7

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.

Week 8

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.

Week 9

a) 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.

b) 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.

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

Week 10

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

The basic salary is entered interactively through the key board.

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

Week 11

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, then program ask the user for the necessary information, such as the file name, new name and so on.

Week 12

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

b) 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.

Week 13

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.

Week 14

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)

Week 15

Review

Text Books

1. Unix concepts and applications, Fourth Edition, Sumitabha Das, TMH
2. Introduction to UNIX & SHELL programming, M.G. Venkatesh Murthy, Pearson Education

Design Thinking Lab

B. Tech II Year I Semester					Dept. of Computer Science and Engineering			
Code	**Category**	**Hours / Week**			**Credits**	**Marks**		
	ESC LAB	L	T	P	C	CIE	SEE	Total
		0	0	2	1	50	50	100

Course Objectives

1. Understand the concepts of design thinking phases.
2. To familiarize the participant with different case studies
3. Apply both critical thinking and design thinking in parallel to solve real time problems.
4. Apply design thinking phases to real time applications.

Course Outcomes

1. Define the phases of design thinking
2. Explore through different real time case studies
3. Experience a hands-on implementation of design thinking to a real time problem
4. Connect design thinking to real time applications.

Week 1

1. Introduction to phases of Design Thinking

Week 2

2. Empathize to identify problem

Week 3

3.Define the Problem

Week 4

4. Ideate the Problem

Week 5

5.Building of Prototype

Week 6

6.Iterations of Prototype

Week 7

7. Iterations of Prototype

Week 8

8.Demonstration of Prototype Model

Week 9

9. Internal Evaluation of Prototype

Week 10

10. Internal Evaluation of Prototype

Week 11

11. Document submission

Week 12 and 13

Review

Reference Books

1.Design & Thinking Documentary, https://nyu.kanopy.com/video/design-and-thinking
2. Stephanie di Russo, Understanding the Behaviour of Design Thinking in Complex Environments,
3.https://www.academia.edu/24919250/Understanding_the_behaviour_of_design_thinking_in_complex_environments

Computer Organization and Architecture