Course Code: 19CS2111

COMPUTER ORGANIZATION

B.Tech. II Year I Semester	**L**	**T**	**P**	**C**
	3	-	-	**3**

Course Objectives:

- To introduce principles of computer organization and the basic architectural concepts.
- To understand the design concepts of control memory and central processing unit.
- To explain the computer arithmetic set of operations and instruction set design.
- To understand the representation of data at the machine level and how computations are performed at machine level.
- To understand the memory organization and I/O organization.
- To become familiar with pipelining, vector processing and memory organization.

Course Outcomes:

- Able to explain the principles of computer organization and simple register transfer language to specify various computer operations.
- Able to compare different addressing modes and instruction formats.
- Able to make use of all computer arithmetic operations.
- Able to decide the type of data representations.
- Able to identify the types of memory organizations.
- Able to compare various types of pipeline and processing.

UNIT - I

Digital Computers: Introduction, Block Diagram of Digital Computer, Definition of Computer Organization, Computer Design and Computer Architecture.

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 Registers, Computer Instructions, Timing and Control, Instruction Cycle, Memory Reference Instructions, Input-Output and Interrupt.

UNIT - II

Micro Programmed Control: Control Memory, Address Sequencing, Micro Program example, Design of Control Unit.

Central Processing Unit: General Register Organization, Instruction Formats, Addressing Modes, Data Transfer and Manipulation, Program Control.

UNIT - III

Data Representation: Data Types, Complements, Fixed Point Representation, Floating Point Representation.

Computer Arithmetic: Addition and Subtraction, Multiplication Algorithms, Division Algorithms, Floating– point Arithmetic Operations. Decimal Arithmetic Unit, Decimal Arithmetic Operations

UNIT - IV

Input-Output Organization: Input-Output Interface, Asynchronous Data Transfer, Modes of Transfer, Priority Interrupt, Direct Memory Access.

Memory Organization: Memory Hierarchy, Main Memory, Auxiliary Memory, Associate Memory, Cache Memory.

UNIT - V

Reduced Instruction Set Computer: CISC Characteristics, RISC Characteristics.

Pipeline and Vector Processing: Parallel Processing, Pipelining, Arithmetic Pipeline, Instruction Pipeline, RISC Pipeline, Vector Processing, Array Processor.

Multi Processors: Characteristics of Multiprocessors, Interconnection Structures, Interprocessor Arbitration, Inter processor communication and synchronization, Cache Coherence.

TEXT BOOK:

[1] Computer System Architecture – M. Morris Mano, 3rd Edition, Pearson/PHI.

REFERENCE BOOKS:

[1] Computer Organization – Carl Hamacher,ZvonkoVranesic, SafwatZaky,5th Edition, McGraw-Hill.

[2] Computer Organization and Architecture – William Stallings 6th Edition, Pearson/PHI.

[3] Structured Computer Organization – Andrew S. Tanenbaum, 4th Edition, PHI/Pearson

Course Code: 19CS2112

OBJECT ORIENTED PROGRAMMING THROUGH JAVA

B.Tech II Year I Semester	**L**	**T**	**P**	**C**
	3		-	**3**

Prerequisites:

- Fundamental knowledge of object-oriented concepts, terminology, and syntax, and the steps required to create basic Java programs.
- Previous experience with at least one programming language

Course Objectives:

- To understand object oriented programming concepts, and apply them in solving problems.
- To introduce the implementation of packages and interfaces.
- To introduce the concepts of exception handling and multithreading.
- To use the collection framework classes in to real time scenarios.
- To introduce the design of Graphical User Interface using applets and swing controls

Course Outcomes:

- Able to solve real world problems using OOP techniques.
- Able to understand the use of abstract classes.
- Able to solve problems using java collection framework and I/O classes.
- Able to develop multithreaded applications with synchronization.
- Able to develop applets for web applications.
- Able to design GUI based applications.

Unit I:

Object Orientated Paradigm:

Introduction to Procedural and Object oriented programming, A way of viewing world – agents and communities, responsibilities, messages, methods, class and instance, Oops concepts.

Java Basics:

Introduction to Java ,JVM Architecture, Java Buzz words, Java Tokens- Comments, Identifiers, Keywords, Separators, Data types, enumerated types, Variables, constants, Type Conversion, Operators, Control Statements, Wrapper Classes, Structure of java with simple standalone program, arrays, console input and output, formatting output, constructors, methods – static and instance, parameter passing, access control modifiers, this reference, overloading methods and constructors, recursion, garbage collection, Inner classes, exploring String and String Buffer class.

Unit II:

Inheritance – Inheritance hierarchies, super and sub classes, super keyword, preventing inheritance: final classes and methods, the Object class and its methods.

Polymorphism- dynamic binding, static binding, abstract classes and methods. Interfaces – Interfaces vs. Abstract classes, defining an interface, Multiple Inheritance through interface, extending interface.
Packages-Defining, Creating and Accessing a Package, Understanding CLASSPATH, importing packages.

UNIT – III
I/O STREAM - Introduction, Byte-oriented streams, Character – oriented streams, File streams, Random-access file, Serialization.
Exception handling -- Dealing with errors, benefits of exception handling, classification of exceptions- exception hierarchy, checked exceptions and unchecked exceptions, usage of try, catches, throw, throws and finally, built in exceptions and user defined exceptions.
Multithreading – Differences between processes and threads, thread life cycle, creating threads, interrupting threads, multithreading ,thread priorities, Synchronizing, inter-thread communication.

Unit IV:
Collection framework in java: Introduction, Util Package interfaces, Retrieving elements from collections.
Collection interfaces: Set, Map, List ,Queue, Implementation classes : HashSet, HashMap, ArrayList, Stack, Linked List, String Tokenizer, Scanner, Calendar class.

Unit V:
AWT - Introduction to AWT, Components, Event, Event-Delegation-Model, Listeners, Layout management and types – border, grid and flow.
Individual components: Lable, Button, Checkbox, Radio Button. Choice, List, Menu, Text Field, Text Area.
Applets - Inheritance hierarchy for applets, differences between applets and applications, life cycle of an Applet, passing parameters to applets,
Swings- Introduction to Swing, Swing vs.AWT, Hierarchy for Swing components

TEXT BOOKS:
[1] Java ;The complete reference,7th edition, Herbert Schildt,TMH
[2] Understanding OOP with java, updated edition, T.Budd, Pearson education
REFERENCE BOOKS:
[1] JAVA Fundamentals- A comprehensive introduction, Herbert Schildt and Dale Skrien, TMH
[2] Java for Programmers, P.J.Deitel and H.M.Deitel, Perason education (OR) JAVA: How to program P.J.Deitel and H.M.Deitrl, PHI
[3] Thinking in java, Bruce Eckel, Pearson Education
[4] Object Oriented Programming through java, P.Radha Krishna, Universities Press.
[5] Programming in java, S.Malhotra and S.Choudhary, Oxford Univ. Press.

Course code 19CS2113

DATA STRUCTURES

B.TECH II Year I Semester. **L T P C**
3 1 0 4

Prerequisites: A course on "Programming for Problem Solving".

Course Objectives:

- Exploring basic data structures such as stacks and queues.
- Introduces a variety of data structures such as hash tables, search trees, tries, heaps ,graphs.
- Introduces searching, sorting and pattern matching algorithms

Course Outcomes:

- Ability to design and analyze the time and space complexity and understand the concepts Linked List of data structures
- Understand data structure concepts of Stacks and Queues
- Understand data structure concepts of Trees, Graphs
- Understand the concepts of Searching and Sorting.
- Understand the concepts of Search Trees.
- Understand the concepts of Pattern Matching and Tries.

UNIT- I

Basic concepts : Algorithm Specification-Introduction, Recursive algorithms, Data Abstraction Performance Analysis- time complexity and space complexity, Asymptotic Notation-Big O, Omega and Theta notations, Introduction to Linear and Non Linear data structures.

Linked Lists : Singly Linked Lists-Operations-Insertion, Deletion, Concatenating singly linked lists, Circularly linked lists- Operations for Circularly linked lists, Doubly Linked Lists-Operations- Insertion, Deletion. Representation of single, two dimensional arrays, sparse matrices-array and linked representations.

UNIT- II

Introduction to Data Structures - Stack ADT, definition, operations, array and linked implementations in C, applications-infix to postfix conversion, Postfix expression evaluation, recursion implementation, Queue ADT, definition and operations, array and linked Implementations in C, Circular Queues -Insertion and deletion operations, Deque (Double ended Queue)ADT, array and linked implementations in C.

UNIT- III

Trees – Terminology, Representation of Trees, Binary tree ADT, Properties of Binary Trees, Binary Tree Representations-array and linked representations, Binary Tree traversals, Threaded binary trees, Max Priority Queue ADT-implementation-Max Heap-Definition, Insertion into a Max Heap, Deletion from a Max Heap.

Graphs – Introduction, Definition, Terminology, Graph ADT, Graph Representations- Adjacency matrix, Adjacency lists, Graph traversals- DFS and BFS.

UNIT- IV

Searching- Linear Search, Binary Search, Static Hashing-Introduction, hash tables, hash functions, Overflow Handling.

Sorting-Insertion Sort, Selection Sort, Radix Sort, Quick sort, Heap Sort, Comparison of Sorting methods.

UNIT- V

Search Trees-Binary Search Trees, Definition, Operations- Searching, Insertion and Deletion, AVL Trees- Definition and Examples, Insertion into an AVL Tree ,B-Trees, Definition, B-Tree of order m, operations-Insertion and Searching, Introduction to Red-Black and Splay Trees(Elementary treatment-only Definitions and Examples), Comparison of Search Trees.

Pattern matching algorithm- The Knuth-Morris-Pratt algorithm, Tries (examples only).

TEXT BOOKS:

1. Fundamentals of Data structures in C, 2nd Edition, E.Horowitz, S.Sahni and Susan Anderson-Freed, Universities Press.
2. Data structures A Programming Approach with C, D.S.Kushwaha and A.K.Misra, PHI.

REFERENCE BOOKS:

1. Data structures: A Pseudocode Approach with C, 2nd edition, R.F.Gilberg And B.A.Forouzan, Cengage Learning.
2. Data structures and Algorithm Analysis in C, 2nd edition, M.A.Weiss, Pearson.
3. Data Structures using C, A.M.Tanenbaum,Y. Langsam, M.J.Augenstein, Pearson.
4. Data structures and Program Design in C, 2nd edition, R.Kruse, C.L.Tondo and B.Leung,Pearson.
5. Data Structures and Algorithms made easy in JAVA, 2nd Edition, Narsimha Karumanchi, CareerMonk Publications.
6. Data Structures using C, R.Thareja, Oxford University Press.
7. Data Structures, S.Lipscutz,Schaum's Outlines, TMH.
8. Data structures using C, A.K.Sharma, 2nd edition, Pearson.
9. Data Structures using C &C++, R.Shukla, Wiley India.
10. Classic Data Structures, D.Samanta, 2nd edition, PHI.
11. Advanced Data structures, Peter Brass, Cambridge

Course code :19BS2115

DISCRETE MATHEMATICAL STRUCTURES

B.TECH II Year I Semester

L	T	P	C
3	-	-	3

Prerequisites: An understanding of Mathematics in general is sufficient.

Course Objectives

- Translate statements from a natural language into its symbolic structures in logic.
- Define the syntax and semantics of propositional and predicate logic.
- To learn set theory, Relations, functions, ordering relations.
- To introduce generating functions and recurrence relations.
- To learn Graph Theory for solving problems.

Course Outcomes:

- Ability to understand and construct precise mathematical proofs.
- Ability to use Mathematical logic to formulate precise statements.
- Ability to perform operations on discrete structures such as sets, functions and relations.
- Ability to solve discrete mathematics problems that involve computing Permutations and combinations of a set.
- Ability to analyze and solve problems involving recurrence relations and generating functions.
- Ability to apply graph theory in solving computing problems**.**

UNIT – I:

Mathematical Logic and Proofs: Propositional Logic, Applications of Propositional Logic, Propositional Equivalence, Predicates and Quantifiers, Nested Quantifiers, Rules of Inference, Introduction to Proofs, Proof Methods and Strategy: Direct Proof, Indirect Proof, and Proof by Contradiction.

UNIT-II:

Sets and Relations: Sets, Functions, Cardinality of Sets, Relations and their Properties, Representing Relations, Matrix Representation of Relations, Closures of Relations, Equivalence Relations, Partial Ordering, Lattices.

Algebraic Structures: Introduction, Algebraic Systems, Semi groups and Monoids, Groups.

UNIT- III:

Elementary Combinatorics : Basics of Counting, Combinations and Permutations, Enumeration of Combinations and Permutations, Enumerating Combinations and Permutations with Repetitions, Enumerating Permutations with Constrained Repetitions, Binomial Coefficients, The Binomial and Multinomial Theorems.

UNIT IV:

Recurrence Relations: Sequences and Summations, Generating Functions, Calculating coefficients of Generating Functions, Recurrence Relations, Solving Linear Recurrence Relations by substitution method and Generating Functions, Inclusion-Exclusion, Applications of Inclusion-Exclusion.

UNIT - V

Graphs: Graphs and Graph Models, Graph Terminology and Special Types of Graphs, Representing Graphs and Graph Isomorphism, Connectivity, Euler and Hamilton Paths, Shortest-Path Problems, Planar Graphs, Chromatic number, Graph Coloring.

Trees: Introduction to Trees, Applications of Trees, Spanning Trees, Minimum Spanning Trees.

TEXT BOOKS:

1. Discrete Mathematics and its Applications with Combinatorics and Graph Theory – Kenneth H Rosen, 7th Edition, TMH.

REFERENCE BOOKS:

1. Discrete Mathematical Structures with Applications to Computer Science – J.P. Tremblay and R. Manohar, TMH.
2. Discrete Mathematics for Computer Scientists and Mathematicians: Joe L. Mott, Abraham Kandel, Teodore P.Baker, 2nd Edition, Pearson Education.
3. Discrete Mathematics- Richard Johnsonbaugh, 7th Edn., Pearson Education.
4. Discrete Mathematics with Graph Theory – Edgar G.Goodaire, Michael M.Parmenter.
5. Discrete and Combinatorial Mathematics – an applied introduction: Ralph. Grimald, 5th edition, Pearson Education.

Course Code: 19EC2117

DIGITAL ELECTRONICS

B.Tech. II Year I Semester.	**L**	**T**	**P**	**C**
	3	**-**	**-**	**3**

Course Objectives:

- To introduce components such as diodes and BJTs.
- To know the applications of components.
- To give understanding of various types of logic families.
- To learn basic techniques for the design of digital circuits and fundamental concepts used in the design of digital systems.
- To understand the concepts of combinational logic circuits and sequential circuits.

Course Outcomes:

- Upon completion of the Course, the students will be able to:
- Know the characteristics of various components.
- Understand the utilization of components.
- Know about the logic families and realization of logic gates.
- Learn Postulates of Boolean algebra and to minimize combinational functions.
- Design and analyze combinational and sequential circuits

UNIT - I

Diodes and Applications: Junction diode characteristics: Open circuited p-n junction, V-I characteristics, effect of temperature, diode resistance, diode capacitance, diode switching times, p-n junction as a rectifier, Tunnel diodes, photo diode, LED.

BJTs: Transistor characteristics: The junction transistor, transistor as an amplifier, CB, CE, CC configurations, comparison of transistor configurations, the operating point, self-bias or Emitter bias, bias compensation, thermal runaway and stability.

UNIT -II

Number Systems: Number systems, Complements of Numbers, Codes- Weighted and Non-weighted codes and its Properties, Parity check code and Hamming code.

Digital Circuits: Digital (binary) operations of a system, OR gate, AND gate, NOT, EXCLUSIVE OR gate, Universal gates, De Morgan Laws, NAND and NOR DTL gates, HTL and TTL gates, RTL and DCTL, Comparison of logic families.

UNIT - III

Combinational Logic Circuits I: Basic Theorems and Properties of Boolean Algebra, Canonical and Standard Forms, Digital Logic Gates, Karnaugh Map Method, Product-of-Sums Simplification, Don’t-Care Conditions, NAND and NOR Implementation.

UNIT -IV

Combinational Logic Circuits II: Binary Adder-Subtractor, Decimal Adder, Binary Multiplier, Magnitude Comparator, Decoders, Encoders, Multiplexers, De multiplexers, Parity generator.

UNIT - V

Sequential Logic Circuits: Sequential Circuits, Storage Elements: Latches and flip flops, Analysis of Clocked Sequential Circuits, State Reduction and Assignment, Shift Registers, Ripple Counters, Synchronous Counters, Random-Access Memory, Read-Only Memory.

TEXTBOOKS

[1] Integrated Electronics: Analog and Digital Circuits and Systems, 2/e, JaccobMillman, Christos Halkias and Chethan D. Parikh, Tata McGraw-Hill Education, India, 2010.

[2] Digital Design, 5/e, Morris Mano and Michael D. Cilette, Pearson, 2011.

REFERENCE BOOKS

[1] Electronic Devices and Circuits, Jimmy J Cathey, Schaum's outline series, 1988.

[2] Digital Principles, 3/e, Roger L. Tokheim, Schaum's outline series, 1994

Course Code : 19CS2151

DATA STRUCTURES LAB

B.Tech II Year I Semester.

L	T	P	C
0	0	3	1.5

Prerequisites: A Course on "Programming for problem solving".

Course Objectives:

- It covers various concepts of C programming language
- It introduces searching and sorting algorithms
- It provides an understanding of data structures such as stacks and queues.

Course Outcomes:

- Ability to develop C programs for computing and real-life applications using basic elements like control statements, arrays, functions, pointers and strings, and data structures like stacks, queues and linked lists.
- Ability to Implement searching and sorting algorithms

LIST OF EXPERIMENTS:

Week 1:

Write a program that uses functions to perform the following operations on Singly Linked List :

i) Creation ii) Insertion iii) Deletion iv) Traversal

Week 2:

Write a program that uses functions to perform the following operations on Doubly Linked List :

i) Creation ii) Insertion iii) Deletion iv) Traversal

Week 3:

Write a program that uses functions to perform the following operations on Circular Linked List. :

i) Creation ii) Insertion iii) Deletion iv) Traversal

Week 4:

Write a program that implement Stack (its operations) using Arrays and Linked list.

Week 5:

Write a C program that uses stack operations to convert a given Infix expression into its Postfix

.

Week 6:

Write a program that implement Queue (its operations) using Arrays and Linked list.

Week 7:

Write a C Program to implement a Double Ended Queue ADT using

i) Arrays ii) Doubly Linked List

Week 8:

Write a C program to implement all the functions of a Dictionary (ADT) using Hashing.

Week 9:

Write a program that use recursive functions to perform the following searching operations for a Key value in a given list of integers:

i) Linear Search ii) Binary Search

Week 10:

Write a program to implement the tree traversal methods.

Week 11:

Write a program to implement the graph traversal methods.

Week 12:

Write a program that implements the following sorting methods to sort a given list of integers in ascending order

i) SelectionSort ii) Quick Sort

Week 13:

Write a program that implements the following sorting methods to sort a given list of integers in ascending order

i) Insertion Sort ii) Merge Sort

Week 14:

Write a program that implements the following sorting methods to sort a given list of integers in ascending order

i) Radix Sort ii) Heap Sort

Week 15:

Write a C program for implementing Knuth-Morris- Pratt pattern matching algorithm.

TEXTBOOKS:

1. Fundamentals of Data Structures in C, 2nd Edition, E. Horowitz, S. Sahni and Susan Anderson Freed, *UniversitiesPress*.
2. DataStructuresusingC–A.S.Tanenbaum,Y.Langsam,andM.J.Augenstein, *PHI/Pearson Education*.

REFERENCE BOOKS:

1. Data Structures: A Pseudocode Approach with C, 2nd Edition, R. F. Gilberg and B. A. Forouzan, CengageLearning.

Course Code:19CS2152

JAVA PROGRAMMING LAB

B.Tech II Year I Semester	**L**	**T**	**P**	**C**
	-	**-**	**3**	**1.5**

Prerequisites:

- Fundamental knowledge of object-oriented concepts, terminology, and syntax, and the steps required to create basic Java programs.
- Previous experience with at least one programming language

Course Objectives:

- To introduce Java compiler and eclipse platform.
- To make the student learn an object oriented way of solving problems using java.
- To make the students to write programs using multithreading concepts and handle exceptions.
- To make the students to write programs that connects to a database and be able to perform various operations.
- To make the students to create the Graphical User Interface using Applets, AWT Components & Swing Components.

Course Outcomes:

- Able to use Java compiler and eclipse platform to write and execute java program.
- Understand and Apply Object oriented features and Java concepts.
- Able to apply the concept of multithreading and implement exception handling.
- Able to access data from a Database with java program.
- Develop applications using Console I/O and File I/O,GUI applications

1**)** Use eclipse or Netbean platform and acquaint with the various menus, create a test project, add a test class and run it see how you can use auto suggestions, auto fill. Try code formatter and code refactoring like renaming variables, methods and classes. Try debug step by step with a small program of about 10 to 15 lines which contains at least one if else condition and a for loop.

2) Write a java program to demonstrate mutability of StringBuffer class to find out whether a given string is Palindrome or not?

3) Write a program to create a user defined package named **'sample'** and demonstrate importing this package in other program.

4**)** Write a Java program that works as a simple calculator. Use a grid layout to arrange buttons for the digits and for the +, -,*, % operations. Add a text field to display the result. Handle any possible exceptions like divide by zero.

5) a) Develop an applet that displays a simple message.
b) Develop an Applet that receives an integer in one text field & compute its factorial value & returns it in another text filed when the button "Compute" is clicked

6) Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a Number Format Exception. If Num2 were Zero, the program would throw an Arithmetic Exception Display the exception in a message dialog box

7) Write a java program that implements a multi-thread application that has three threads. First thread generates random integer every 1 second and if the value is even, second thread computes the square of the number and prints. If the value is odd, the third thread will print the value of cube of the number.

8) Write a java program that simulates a traffic light. The program lets the user select one of three lights: red, yellow, or green with radio buttons. On selecting a button, an appropriate message with "stop" or "ready" or "go" should appear above the buttons in a selected color. Initially there is no message shown.

9) Write a java program to create an abstract class and abstract methods

10) Suppose that a table named Table.txt is stored in a text file. The first line in the file header and the remaining lines correspond to row in the table. The elements are separated by commas. Write a Java program to display the table using labels in grid layout.

11) Write a Java program that handles all mouse events and shows the event name at the center of the window when a mouse event is fired. (Use adapter classes).

12) Write a java program that loads names and phone numbers from a text file where the data is organized as one line per record and each field in a record are separated by a tab (\t).it takes a name or phone number as input and prints the corresponding other value from the hash table(hint: use hash tables)

13) Write a java program that correctly implements the producer–consumer problem using the concept of inter thread communication.

14) Write a java program to list all the files in a directory including the files present in all its subdirectories.

TEXT BOOKS:

[1] Java how to program,sixthedition,H.M.Ditiel
[2] Programming with java,M.P.Bhave

Course Code: 19EC2155

DIGITAL ELECTRONICS LAB

B.Tech. II Year I Sem.	**L**	**T**	**P**	**C**
	-	**-**	**2**	**1**

Course Objectives:

- To introduce components such as diodes, BJTs.
- To know the applications of components.
- To learn basic techniques for the design of digital circuits
- To learn basic fundamental concepts used in the design of digital systems.
- To understand the concepts of sequential circuits.

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

- Know the characteristics of various components.
- Understand the utilization of components.
- Understand the Postulates of Boolean algebra.
- Known about the logic families and realization of logic gates.
- Know the minimize combinational functions
- Design and analyze sequential circuits

LIST OF EXPERIMENTS

1. PN Junction diode characteristics A) Forward bias B) Reverse bias.

2. Half Wave Rectifier without filters.

3. Full Wave Rectifier without filters.

4. Switching characteristics of a transistor.

5. Input and output characteristics of BJT in CE Configuration.

6. Input and output characteristics of BJT in CB Configuration.

7. Realization of Boolean Expressions using Gates.

8. Realization of logic gates using DTL, TTL, RTL etc.

9. Design and realization logic gates using universal gates.

10. Generation of clock using NAND / NOR gates.

11. Design a 4 – bit Adder / Subtractor.

12. Design and realization a Synchronous and Asynchronous counter using flip-flops.

TEXT BOOKS

[1] Integrated Electronics: Analog and Digital Circuits and Systems, 2/e, JaccobMillman, Christos Halkias and Chethan D. Parikh, Tata McGraw-Hill Education, India, 2010.
[2] Digital Design, 5/e, Morris Mano and Michael D. Cilette, Pearson, 2011.

REFERENCE BOOKS

[1] Electronic Devices and Circuits, Jimmy J Cathey, Schaum's outline series, 1988.
[2] Digital Principles, 3/e, Roger L. Tokheim, Schaum's outline series, 1994

Course Code: 19CS2211

OPERATING SYSTEMS

B.Tech. II Year II Semester.	**L**	**T**	**P**	**C**
	3	**-**	**-**	**3**

Course Objectives:

- Provide an introduction to operating system concepts (i.e, Operating System services, OS Structure)
- Introduces the way an ***operating system*** can make the computer more productive by the effective **management of processes** (i.e., threads, scheduling, Synchronization)
- Introduce basic UNIX commands, system call interface for process management; inter process communication and I/O in UNIX.
- To understand the basic **memory management** of operating system.
- To elucidate **deadlocks**, present a number of various techniques for preventing or avoiding or recovering from deadlocks in a computer system
- Introduce the issues to be considered in the design and development of operating system with high **protection** and ease of access.

Course Outcomes:

- Will be able to explain services & structure of operating systems.
- Demonstrate the knowledge of the components of computer and their respective roles in computing and illustrate various methods of process scheduling ,synchronization.
- Ability to recognize and resolve user problems related to memory management with standard operating system techniques.
- Gain practical knowledge of how programming languages, operating systems, and architectures interact and how to use each effectively to implement file system directory Structures.
- Will be able to apply security mechanisms and techniques to handle deadlocks.
- Will be able to do Programming and debugging C code at the system level communicating directly with an operating system via system calls.

UNIT - I

Overview-Introduction: Operating system objectives, User view, System view, Operating system definition ,Computer System Organization, Computer System Architecture, OS Structure, OS Operations, Process Management, Memory Management, Storage Management, Protection and Security, Computing Environments. Operating System services, User and OS Interface, System Calls, Types of System Calls, System Programs, Operating System Design and Implementation, OS Structure.

UNIT - II

Process and CPU Scheduling: Process concepts-The Process, Process State, Process Control Block, Threads, Process Scheduling-Scheduling Queues, Schedulers, Context Switch,

Operations on Processes, System calls-fork(),exec(),wait(),exit(),. Process Scheduling-Basic concepts, Scheduling Criteria, Scheduling algorithms, Multiple Processor Scheduling, Real-Time Scheduling, Thread scheduling, Linux scheduling and Windows scheduling.

Inter process communication: Background, IPC using ordinary pipes and named pipes in Unix.

Process Synchronization: Background, The Critical Section Problem, Peterson's solution, Synchronization Hardware, Semaphores, Classic Problems of Synchronization, Monitors, Synchronization in Linux and Windows.

UNIT - III

Memory Management: Background, Swapping, Contiguous Memory Allocation, Segmentation, Paging, Structure of Page Table.

Virtual Memory Management: Background, Demand Paging, Copy-on-Write, Page Replacement, Page Replacement Algorithms, Allocation of Frames, Thrashing, Virtual memory in Windows.

UNIT - IV

Storage Management-File System: Concept of a File, System calls for file operations-**o**pen(), read(), write(), close(), seek(), unlink(), Access methods, Directory, File System Mounting, File Sharing, Protection.

File System Implementation: File System Structure, File System Implementation, Directory Implementation, Allocation methods, Free-space Management, Efficiency, and Performance. **Mass Storage Structure**: Overview of Mass Storage Structure, Disk Structure, Disk Scheduling, Disk Management, Swap space Management.

UNIT - V

Deadlocks: System Model, Deadlock Characterization, Methods for Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, and Recovery from Deadlock. **Protection**: System Protection, Goals of Protection, Principles of Protection, Domain of Protection, Access Matrix, Implementation of Access Matrix, Access Control, Revocation of Access Rights.

TEXT BOOKS

[1] Operating System Principles- Abraham Silberchatz, Peter B. Galvin, Greg Gagne 7th Edition, John Wiley

[2] Advanced programming in the UNIX environment, W.R. Stevens, Pearson education.

REFERENCE BOOKS

[1] Operating Systems – Internals and Design Principles Stallings, Fifth Edition–2005, Pearson Education/PHI

[2] Operating System A Design Approach- Crowley, TMH.

[3] Modern Operating Systems, Andrew S. Tanenbaum 2nd edition, Pearson/PHI

[4] UNIX programming environment, Kernighan and Pike, PHI/ Pearson Education

[5] UNIX Internals -The New Frontiers, U. Vahalia, Pearson Education.

Course Code: 19CS2212

DATABASE MANAGEMENT SYSTEMS

B.Tech. II Year II Sem.	**L**	**T**	**P**	**C**
	3	**-**	**-**	**3**

Course Objectives:

- To understand the basic database concepts, applications, data models, schemas and instances.
- To familiarize Entity Relationship model for a database.
- To demonstrate the use of constraints and relational algebra operations.
- To become proficient in the basics of SQL and construct queries using SQL.
- To become familiar in the normalization techniques to organize data in databases.
- To demonstrate the basic concepts of transaction processing and concurrency control.
- To familiarize the concepts of database storage structures and the access techniques

Course Outcomes:

- Demonstrate the basic elements of a relational database management system.
- Ability to design entity relationship model and convert in to relational model.
- Formulate SQL queries on the data.
- Apply normalization for the development of application software.
- Analyze transaction processing, concurrency control and recovery management techniques.
- Analyze the storage structures and indexing.

UNIT – I

Introduction: Database system applications, Purpose of data base systems, Data Independence, Data Abstraction- View of data, Database System architecture, data models, schema and instances,

Database Design Process, ER Diagrams - Entities, Attributes, Relationships, Constraints, keys, additional features of E-R model, Conceptual design with the E-R model.

UNIT - II

The Relational Model: Introduction to the relational model, Integrity constraints over relations, Enforcing integrity constraints, querying relational data,

Logical database design: E-R to relational, Introduction to views, Destroying/altering tables and views.

Relational Algebra and Calculus: relational algebra operators, relational calculus - Tuple and domain relational calculus,

SQL: Forms of Basic SQL Query-examples, aggregate functions, Built-in functions, set comparison operators, nested queries, correlated queries, group by, having, order by, joins. Transaction control commands, cursors, stored procedures, Triggers.

UNIT – III

Schema Refinement and Normal Forms: Introduction to schema refinement, functional dependencies, reasoning about FDs. Normal forms - 1NF, 2NF, 3NF, BCNF, properties of decomposition, normalization, schema refinement in database design, other kinds of dependencies- 4 NF,5NF.

UNIT - IV

Transaction Management: Transaction concept, transaction state, implementation of atomicity and durability, concurrent executions, Serializability, recoverability, implementation of isolation.

Concurrency Control and Recovery System: Concurrency control, lock based protocols, time-stamp based protocols, validation based protocols, multiple granularity.

Recovery system - failure classification, storage structure, recovery and atomicity, log based recovery, shadow paging, recovery with concurrent transactions.

UNIT – V

Overview of External Storage : Data on External Storage, File Organization and Indexing, Cluster Indexes, Primary and Secondary Indexes, Index data Structures, Hash Based Indexing, Tree base Indexing, Comparison of File Organizations, Indexes and Performance Tuning

Overview of Indexing: Tree structured indexing - intuition for tree indexes, indexed sequential access method (ISAM), B+ Trees .

TEXT BOOKS

[1] Data base Management Systems, Raghu Ramakrishnan, Johannes Gehrke, McGraw Hill Education (India) Private Limited, 3rd Edition. (Part of UNIT-I, UNIT-II, UNIT-III, UNIT-V)

[2] Data base System Concepts, A. Silberschatz, Henry. F. Korth, S. Sudarshan, McGraw Hill Education(India) Private Limited l, 6th edition.(Part of UNIT-I, UNIT-IV)

REFERENCE BOOKS

[1] Database Systems, 6th edition, R Elmasri, Shamkant B.Navathe, Pearson Education..

[2] Introduction to Database Systems, C.J.Date Pearson Education

[3] Database System Concepts, Peter Rob & Carlos Coronel, Cengage Learning.

[4] Introduction to Database Management, M. L. Gillenson and others, Wiley Student Edition.

Course Code: 19CS2213

FORMAL LANGUAGES AND AUTOMATA THEORY

B.Tech II Year II Semester	**L**	**T**	**P**	**C**
	3	**1**	**-**	**4**

Course Objectives:

- To introduce Formal Languages, Automata Theory and Abstract models of Computation and Computability, Computational complexities and NP – Completeness.
- To gain knowledge in computational theory.
- Explain the theoretical functions of computer science concerning the relationships between languages and machines, the inherent limits of what can be computed and inherent efficiency of solving problems.
- To realize the theoretical concepts and techniques involved in the software system development.
- Build the foundation for students to pursue research in the areas of Automata Theory, Formal Languages and Computational power of machines.

Course Outcomes:

- Acquire a fundamental understanding of the core concepts in automata theory and formal languages.
- An ability to design grammars and automata (recognizers) for different language classes.
- Apply the theoretical concepts and techniques in designing the software systems.
- An ability to identify formal language classes and prove language membership properties.
- An ability to prove and disprove theorems establishing key properties of formal languages and automata.
- Acquire a fundamental understanding of core concepts relating to the theory of computation and computational models including (but not limited to) decidability and intractability.

UNIT - I

Automata: Strings, Alphabet, Language, Operations, Finite State Machine, definitions, finite automation model, acceptance of strings and languages, Deterministic finite automation, Non deterministic Finite automata, Equivalence between NFA and DFA, Conversion of NFA to DFA, minimization of FSM, equivalence between two FSM's, Finite automata with Epsilon transitions, Moore and Malay machines, Equivalence between Mealy and Moore machines.

UNIT - II

Regular Expressions: Regular sets, regular expressions, identity rules, Algebraic Laws for Regular Expressions, Applications of Regular Expressions, equivalence between RE and FA, inter conversion between RE and FA, Arden's theorem, Pumping lemma, Closure properties of regular sets(proofs not required), Decision Properties of Regular Languages, regular grammars, right linear and left linear grammars equivalence between regular linear grammar and FA, inter conversion between RE and RG.

UNIT - III

Context Free Grammars: Context free Grammars, Derivation trees, Left Most Derivations, Right Most Derivations, Ambiguity in Context-Free Grammars, Specifications of Context Free Grammars, Normal Forms, Chomsky Normal Form (CNF), Greibach Normal Form (GNF), and Applications of Context-Free Grammars.

Pushdown Automata: Definition of the Pushdown Automaton, the Languages of a PDA, Equivalence of PDA's and CFG's, Deterministic Pushdown Automata.

UNIT - IV

Turing Machine: Turing machine, definition, model, design of TM, Computable Functions, recursive enumerable language, Church's Hypothesis, Counter machine, types of TM's(Proofs not required).

UNIT- V

Classes of Problems: Chomsky hierarchy of languages, linear bounded automata and context sensitive language, Introduction to DCFL and DPDA,LR(O) Grammar, decidability of problems, Universal Turing Machine, post correspondence problem. Turing reducibility, definition of P and NP problems, NP complete and NP hard problems.

TEXT BOOKS

[1] Hopcroft, John E.; Motwani, Rajeev; Ullman, Jeffrey D. (2013). Introduction to Automata Theory, Languages, and Computation (3rd Ed.). Pearson. ISBN 1292039051

[2] Kamala Krithivasan and Rama. R, "Introduction to Formal Languages, Automata Theory and Computation", Pearson Education 2009

[3] Theory of Computer Science – Automata languages and computation, Mishra and Chandrashekaran, 2nd edition, PHI.

REFERENCE BOOKS

[1] John C Martin, "Introduction to Languages and the Theory of Computation", Third Edition, Tata McGraw Hill Publishing Company, New Delhi, 2007

[2] Introduction to the Theory of Computation, Michael Sipser, 3rd edition, Cengage Learning

[3] Introduction to Computer Theory, Daniel I.A. Cohen, John Wiley.

Course Code: 19CS2214

DESIGN AND ANALYSIS OF ALGORITHMS

B.Tech. II Year II Semester.

L	T	P	C
3	-	-	3

Course Objectives:

- To analyze performance of algorithms.
- To understand and choose the appropriate algorithm design technique for a specified application.
- To solve problems using algorithm design techniques such as the greedy method, divide and conquer, dynamic programming, backtracking and branch and bound.
- To analyze the impact of algorithm design techniques on each application solved.
- To introduce and understand P and NP classes

Course Outcomes:

- Able to analyze the different algorithm design techniques for a given problem.
- Able to design algorithms for various computing problems.
- Able to argue the correctness of algorithms using inductive proofs and invariants.
- Able to synthesize set operations
- Able to explain about coping with the limitations of algorithms.

UNIT - I

Notation of an Algorithm: Fundamentals of Algorithmic Problem Solving, Fundamentals of the Analysis of Algorithm Efficiency–Order Notations and its properties , Mathematical analysis for Recursive -Towers of Hanoi and Non-recursive algorithms, Randomized Algorithms-Monte Carlo and Las Vegas, Amortized analysis Examples.

Divide and conquer- General method-Control abstraction, Solving Recurrence Relation using Substitution method and Master's Theorem, applications - Binary search, Merge sort, Quick sort, Strassen's Matrix Multiplication, Finding Maximum and Minimum element.

UNIT - II

Disjoint Set Operations : Union and find algorithms, AND/OR graphs, Graph traversals-Breadth first search, Depth First search, Connected Components and Spanning trees, Bi-connected components.

Greedy method- General method-Control abstraction, applications- Knapsack problem, Job sequencing with deadlines, Minimum cost spanning trees, Single source shortest path problem.

UNIT - III

Dynamic Programming: General Method, applications-Multi Stage Graphs, Chained matrix multiplication, All pairs shortest path problem, Optimal binary search trees, 0/1 knapsack problem, Reliability design, Traveling sales person problem.

UNIT - IV

Backtracking: General method-Control abstraction, applications-The 8-queen problem, sum of subsets problem, graph coloring, Hamiltonian cycles.

Branch and Bound: General Method-Control abstraction, applications-15-Puzzle Problem-LC search, 0/1 Knapsack problem-LC Branch and Bound solution, FIFO Branch and Bound solution, Traveling sales person problem.

UNIT -V

NP-Hard and NP-Complete problems: Basic concepts, Non-deterministic algorithms, NP – Hard and NP- Complete classes, Cook's theorem- proof of reduction.

Approximation Algorithms for NP – Hard Problems: Traveling Salesman problem, Knapsack problem.

TEXT BOOKS

[1] Ellis Horowitz, SatrajSahni and S Rajasekharam, Fundamentals of Computer Algorithms, Galgotia publishers

[2] M.T. Goodrich, Robert Tamassia, Algorithm design: Foundations, Analysis and Internet examples, Wiley student Edn, John Wiley &sons.

[3] ParagHimanshu Dave, HimanshuBhalchandraDave, Design and Analysis algorithms Pearson Publication.

REFERENCE BOOKS

[1] Allen Weiss, Data structures and Algorithm Analysis in C++, 2nd Edn, Pearson Education

[2] Thomas H.Cormen, Charles E.Leiserson, Ronald L. Rivest and Clifford Stein, "Introduction to Algorithms", Third Edition, PHI Learning Private Limited.

[3] Alfred V. Aho, John E. Hopcroft and Jeffrey D. Ullman, "Data Structures and Algorithms", Pearson Education.

Course Code: 19BS2214

COMPUTER ORIENTED STATISTICAL METHODS

	L	T	P	C
B.TECH II Semester II	**3**	**0**	**0**	**3**

Pre-requisites: Mathematics courses of first year of study.
Course Objectives: To learn

- The theory of Probability, and probability distributions of single and multiple random variables
- The sampling theory and testing of hypothesis and making inferences
- Stochastic process and Markov chains.

Course Outcomes: After learning the contents of this paper the student must be able to

- Apply the concepts of probability and distributions to some case studies
- Correlate the material of one unit to the material in other units
- Resolve the potential misconceptions and hazards in each topic of study.

UNIT - I Probability: Sample Space, Events, Counting Sample Points, Probability of an Event, Additive Rules, Conditional Probability, Independence, and the Product Rule, Bayes' Rule.
Random Variables and Probability Distributions: Concept of a Random Variable, Discrete Probability Distributions, Continuous Probability Distributions, Statistical Independence.
Mathematical Expectation: Mean of a Random Variable, Variance and Covariance of Random Variables.

Unit- II: Discrete Distributions: Bernoulli, Binomial, Geometric Distributions and Poisson distribution.
Continuous Distribution: Continuous Uniform Distribution, Normal Distribution, Areas under the Normal Curve, Applications of the Normal Distribution, Gamma and Exponential Distributions.

Unit- III: Estimation & Test of Hypothesis: Concept of Point estimation and its properties (definition only), Concept of interval estimation with examples. Null & Alternative Hypothesis, Critical region, Type I and Type II errors, level of significance, one tail, two-tail tests. Large sample test for single proportion, difference of proportions, single mean, difference of means

Unit- IV: Small Sample tests: t-Test for single mean, difference of means, paired t-test, F-test.
ANOVA: Introduction, ANOVA for one-way classification only.

UNIT - V Stochastic Processes and Markov Chains: Introduction to Stochastic processes- Markov process. Transition Probability, Transition Probability Matrix, First order and Higher order Markov process, nstep transition probabilities, Markov chain, Steady state condition, Markov analysis.

TEXT BOOKS:

1. Ronald E. Walpole, Raymond H. Myers, Sharon L. Myers, Keying Ye, Probability & Statistics for Engineers & Scientists, 9th Ed. Pearson Publishers.
2. S C Gupta and V K Kapoor, Fundamentals of Mathematical statistics, Khanna publications.
3. S. D. Sharma, Operations Research, Kedarnath and Ramnath Publishers, Meerut, Delhi

REFERENCE BOOKS:

1. T.T. Soong, Fundamentals of Probability And Statistics For Engineers, John Wiley & Sons Ltd, 2004. 2. Sheldon M Ross, Probability and statistics for Engineers and scientists, Academic Press.

Course Code: 19CS2251

DATABASE MANAGEMENT SYSTEMS LAB

B.Tech II Year II Semester	**L**	**T**	**P**	**C**
	-	**-**	**3**	**1.5**

Course Objectives:

Course Objectives:
This lab enables the students to practice the concepts learnt in the subject DBMS by developing a database for any case study like example named "Banking Enterprise".

The student is expected to practice the designing, developing and querying a database in the context of example database "Banking Enterprise". Students are expected to use "Mysql" database.

Course Outcomes:
Ability to design and built an database model for a given case study.
Ability to implement a database schema for a given problem domain
Apply the normalization techniques for development of application software to realistic problems.
Ability to formulate queries using SQL DML/DDL/DCL commands.
Ability to Practice various triggers, procedures, and cursors using PL/SQL.

Experiment 1:
Student should decide on a case study and formulate the problem statement.
Experiment 2:
Conceptual Designing using ER Diagrams (Identifying entities, attributes, keys and relationships between entities, cardinalities, generalization, specialization etc.)
Note: Student is required to submit a document by drawing an ER Diagram for a given problem.
Experiment 3
Converting the above ER Model in to Relational Model (Represent entities and relationships in Tabular form, represent attributes as columns, identifying keys)
Note: Student is required to submit a document showing the database tables created for a given problem domain.
Experiment 4
Normalization -To remove the redundancies and anomalies in the above relational tables, Normalize up to Third Normal Form
Experiment 5
Creation of Tables using SQL- Overview of using SQL tool, Data types in SQL, Creating Tables (along with Primary and Foreign keys), Altering Tables and Dropping Tables
Experiment 6
Practicing DML commands- Insert, Select, Update, Delete.

Experiment 7
Practicing Queries using ANY, ALL, IN, EXISTS, NOT EXISTS, UNION, INTERSECT, CONSTRAINTS etc.

Experiment 8
Formulate Sub queries (Nested, Correlated) and Joins (Inner, Outer and Equi).

Experiment 9
Queries using COUNT, SUM, AVG, MAX, MIN, GROUP BY, HAVING, Creating and Dropping views.

Experiment 10:
PL/SQL programs using conditional statements and loops.

Experiment 11
Triggers - creation of trigger, Insertion using trigger, Deletion using trigger, Updating using trigger

Experiment 12
PL/SQL programs using procedures.

Experiment 13
PL/SQL programs on Cursors- Declaring Cursor, Opening Cursor, Fetching the data, closing the cursor.

Text Books:

1. Introduction to SQL, Rick F.vander Lans, Pearson education.
2. Oracle PL/SQL, B.Rosenzweig and E.Silvestrova, Pearson education.
3. Oracle PL/SQL Programming, Steven Feuerstein, SPD.
4. SQL & PL/SQL for Oracle 10g, Black Book, Dr. P.S. Deshpande, Dream Tech.

Reference Books:

1. Oracle Database 11g PL/SQL Programming, M.Mc Laughlin, TMH.
2. SQL Fundamentals, J.J. Patrick, Pearson Education

Course code: 19CS2252

OPERATING SYSTEMS LAB

	L	T	P	C
B.Tech. II Year II Semester.	-	-	2	1

Course Objectives:

- To provide an understanding of the design aspects of operating system concepts through simulation.
- Introduce basic Unix commands, system call interface for process management, inter process communication and I/O in Unix.

Course Outcomes:

- Able to Simulate and implement operating system concepts such as scheduling, deadlock management, file management and memory management.
- Able to implement C programs using UNIX system calls.

Week 1: Write a C program to copy the contents of one file to another using system calls.

Week 2: Write C programs to simulate the following CPU scheduling algorithms:

a) FCFS b) SJF

Week 3: Write C programs to simulate the following CPU scheduling algorithms:

a) Round Robin b) Priority

Week 4: Write a C program to implement the ls | sort command. (Use unnamed Pipe)

Week 5:
Write C programs to implement IPC between two unrelated processes using named pipe.

Week 6: Write a C program to solve the Dining- Philosopher problem using semaphores.

Week 7: Write C programs to simulate the following techniques of memory management:

a) Paging b) Segmentation

Week 8: Write C programs to simulate the following page replacement algorithms:

a) FIFO b) LRU c) LFU

Week 9: Write C programs to simulate the following File allocation methods:

a) Contiguous b) Linked c) Indexed

Week 10: Write C programs to simulate the following File organization techniques:

a) Single level directory b) Two level c) Hierarchical

Week 11:

Write a C program to simulate Bankers Algorithm for Dead Lock Avoidance.

Week 12:

Write a C program to simulate Bankers Algorithm for Dead Lock Prevention.

TEXT BOOKS

[1] Abraham Silberschatz Peter B. Galvin and Greg Gagne, Operating System Concepts, Wiley 8th Edition, 2008.

[2] Advanced Programming in the UNIX Environment by W. Richard Stevens Pearson Education.

[3] Andrew S. Tanenbaum and Herbert Bros, Modern Operating Systems (4th Edition), Pearson

REFERENCE BOOKS

[1] An Introduction to Operating Systems, P.C.P Bhatt, 2nd edition, PHI.

[2] UNIX System Programming Using C++, Terrence Chan, PHI/Pearson.

[3] Modern Operating Systems, Andrew S Tanenbaum, 3rd Edition, PHI.

[4] Sumitabha Das , UNIX Concepts and Applications, Tata McGraw-Hill

Course Code: 19CS2253

DESIGN AND ANALYSIS OF ALGORITHMS LAB

B.Tech. II Year II Sem.	**L**	**T**	**P**	**C**
	-	**-**	**3**	**1.5**

Course Objectives:

- To write programs in java to solve problems using divide and conquer strategy.
- To write programs in java to solve problems using backtracking strategy.
- To write programs in java to solve problems using greedy and dynamic programming techniques.

Course Outcomes:

- Able to write programs in java to solve problems using algorithm design techniques such as Divide and Conquer, Greedy, Dynamic programming, and Backtracking.

Week 1: Write a java program to implement Merge Sort algorithm for sorting a list of integers in ascending order.

Week 2: Write a java program to implement Quick Sort algorithm for sorting a list of integers in ascending order.

Week 3: Write a java program to implement the Depth First Search (DFS) algorithm for a graph.

Week 4: Write a. java program to implement the Breadth First Search (BFS) algorithm for a graph.

Week 5: Write a java program to implement greedy algorithm for job sequencing with deadlines.

Week 6: Write a java program to implement Dijkstra's algorithm for the Single source shortest path problem.

Week 7: Write a java program that implements Prim's algorithm to generate minimum cost spanning tree.

Week 8: Write a java program that implements Kruskal's algorithm to generate minimum cost spanning tree.

Week 9: Write a java program to implement Dynamic Programming algorithm for the 0/1 Knapsack problem.

Week 10: Write a java program to implement Dynamic Programming algorithm for the Optimal Binary Search Tree Problem.

Week 11: Write a java program to implement Floyd's algorithm for the all pairs shortest path problem.

Week 12: Write a java programs to implement backtracking algorithm for the N-queens problem.

Week 13: Write a java program to implement the backtracking algorithm for the sum of subsets problem.

Week 14: Write a java program to implement the backtracking algorithm for the Hamiltonian Circuits problem.

Week 15: Write a java program to Implement Graph Coloring using Back Tracking.

TEXT BOOKS

[1] Data structures, Algorithms and Applications in java, 2nd Edition, S. Sahani, Universities Press.
[2] Data structures and Algorithms in java, 3rd edition, A. Drozdek, Cengage Learning.
[3] Data structures with Java, J. R. Hubbard, 2nd edition, Schaum's Outlines, TMH.

REFERENCE BOOKS

[1] Data structures and algorithms in Java, 2nd Edition, R. Lafore, Pearson Education.
[2] Data Structures using Java, D. S. Malik and P.S. Nair, Cengage Learning

Course Code: 19MC0001

GENDER SENSITIZATION

(An Activity-based Course)

B.Tech. II Year II Semester.	**L**	**T**	**P**	**C**
	2	**-**	**-**	**0**

Course Objectives:

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

Course Outcomes:

- Students will have developed a better understanding of important issues related to gender in contemporary India.
- Students 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.
- Students will attain a finer grasp of how gender discrimination works in our society and how to counter it.
- Students will acquire insight into the gendered division of labour and its relation to politics and economics.
- Men and women students and professionals will be better equipped to work and live together as equals.
- Students will develop a sense of appreciation of women in all walks of life.
- 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: Introduction- Definition of Gender-Basic Gender Concepts and Terminology-Exploring Attitudes towards Gender-Construction of Gender-Socialization: Making Women, Making Men - Preparing for Womanhood. Growing up Male. First lessons in Caste.

UNIT - II

Gender Roles and Relations: Two or Many? -Struggles with Discrimination-Gender Roles and Relations-Types of Gender Roles-Gender Roles and Relationships Matrix-Missing Women-Sex

Selection and Its Consequences-Declining Sex Ratio. Demographic Consequences-Gender Spectrum: Beyond the Binary.

UNIT - III

Gender and Labour: Division and Valuation of Labour-Housework: The Invisible Labor- "My Mother doesn't Work." "Share the Load."-Work: Its Politics and Economics -Fact and Fiction. Unrecognized and unaccounted work-Gender Development Issues-Gender, Governance and Sustainable Development-Gender and Human Rights-Gender and Mainstreaming

UNIT - IV

Gender Based Violence: The Concept of Violence- Types of Gender-based Violence-Gender-based Violence from a Human Rights Perspective-Sexual Harassment: Say No! -Sexual Harassment, not Eve-teasing- Coping with Everyday Harassment- Further Reading: "*Chupulu".* Domestic Violence: Speaking Out Is Home a Safe Place? -When Women Unite [Film]. Rebuilding Lives. Thinking about Sexual Violence Blaming the Victim-"I Fought for my Life…."

UNIT - V

Gender and Culture: Gender and Film-Gender and Electronic Media-Gender and Advertisement-Gender and Popular Literature- Gender Development Issues-Gender Issues-Gender Sensitive Language-Gender and Popular Literature - Just Relationships: Being Together as Equals Mary Kom and Onler. Love and Acid just do not Mix. Love Letters. Mothers and Fathers. Rosa Parks-The Brave Heart.

TEXT BOOKS

[1] All the five Units in the Textbook, "Towards a World of Equals: A Bilingual Textbook on
Gender" written by A.Suneetha, Uma Bhrugubanda, Duggirala Vasanta, Rama Melkote, Vasudha Nagaraj, Asma Rasheed, Gogu Shyamala, Deepa Sreenivas and Susie Tharu and published by Telugu Akademi, Hyderabad,Telangana State in the year 2015.

REFERENCE BOOKS

[1] Menon, Nivedita. Seeing like a Feminist. New Delhi: Zubaan-Penguin Books, 2012
[2] Abdulali Sohaila. "I Fought For My Life…and Won."Available online at:
[3] http://www.thealternative.in/lifestyle/i-fought-for-my-lifeand-won-sohaila-abdulal/