Counselling Code : VBIT

VIGNANA BHARATHI Institute of Technology

(A UGC Autonomous Institution, Approved by AICTE, Accredited by NBA & NAAC-A Grade, Affiliated to JNTUH)

B.Tech Computer Science & Engineering
(Artificial Intelligence and Machine Learning)
R22-COURSE STRUCTURE & SYLLABUS

II YEAR I SEMESTER

S. No.	Course Code	Course Title	Category	L	T	P	Credits
1	22BS2115	Mathematical and Statistical Foundations	BS	3	1	0	4
2	22CS2111	Data Structures	PC	3	0	0	3
3	22CS2112	Computer Organization and Architecture	PC	3	0	0	3
4	22AM2111	Software Engineering	PC	3	0	0	3
5	22AM2112	Operating Systems	PC	3	0	0	3
6	22AM2151	Introduction to Data Structures Lab	PC	0	0	2	1
7	22AM2152	Operating Systems Lab	PC	0	0	2	1
8	22AM2153	Software Engineering Lab	PC	0	0	2	1
9	22AM2154	Node JS/React JS/Django	PC	0	0	2	1
10	22MC0003	Constitution of India	MC	3	0	0	0
		Total		**18**	**1**	**8**	**20**

II YEAR II SEMESTER

S. No.	Course Code	Course Title	Category	L	T	P	Credits
1	22BS2211	Discrete Mathematics	BS	3	0	0	3
2	22AM2211	Automata Theory and Compiler Design	PC	3	0	0	3
3	22CS2212	Database Management Systems	PC	3	0	0	3
4	22AM2212	Introduction to Artificial Intelligence	PC	3	0	0	3
5	22AM2213	Object Oriented Programming through Java	PC	3	0	0	3
6	22AM2251	Java Programming Lab	PC	0	0	2	1
7	22CS2252	Database Management Systems Lab	PC	0	0	2	1
8	22AM2281	Real-time Research Project/Field-Based Research	PW	0	0	4	2
9	22AM2252	Prolog/Lisp/ Pyswip	PC	0	0	2	1
10	22MC0004	Gender Sensitization Lab	MC	0	0	2	0
		Total		**15**	**0**	**12**	**20**

22BS2115: MATHEMATICAL AND STATISTICAL FOUNDATIONS

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

3 1 0 4

Prerequisites: Mathematics courses of first year of study.

Course Objectives:

- The Number Theory basic concepts useful for cryptography etc
- The theory ofProbability,andprobabilitydistributionsofsingleandmultiplerandomvariables
- To acquire knowledge about continuous probability distributions and sampling.
- The sampling theory and testing of hypothesis and making inferences
- Stochastic process and Markov chains.

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

- Apply the number theory concepts to cryptography domain.
- Derive relationship among various performance measures using continuous probability distributions and sampling techniques.
- 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

Greatest Common Divisors and Prime Factorization: Greatest common divisors, The Euclidean algorithm, The fundamental theorem of arithmetic, Factorization of integers and the Fermat numbers **Congruences:** Introduction to congruences, Linear congruences, The Chinese remainder theorem, Systems of linear congruences

UNIT-II

Simple Linear Regression and Correlation: Introduction to Linear Regression, The Simple Linear Regression Model, Least Squares and the Fitted Model, Properties of the Least Squares Estimators, Inferences Concerning the Regression Coefficients, Prediction, Simple Linear Regression Case Study **Random Variables and Probability Distributions:** Concept of a Random Variable, Discrete Probability Distributions, Continuous Probability Distributions, Statistical Independence.

Discrete Probability Distributions: Binomial Distribution, Poisson distribution.

UNIT-III

Continuous Probability Distributions: Normal Distribution ,Areas under the Norma lCurve, Applications of the Normal Distribution, Normal Approximation to the Binomial

Fundamental Sampling Distributions: Random Sampling, Sampling Distributions, Sampling Distribution of Means and the Central Limit Theorem, Sampling Distribution of S2, t–Distribution, F-Distribution.

UNIT-IV

Estimation & Tests of Hypotheses: Introduction, Statistical Inference, Classical Methods of Estimation. Estimating the Mean, Standard Error of a Point Estimate, Prediction Intervals, Tolerance Limits, Estimating the Variance, Estimating a Proportion for single mean, Difference between Two Means, between Two Proportions for Two Samples and Maximum Likelihood Estimation.

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. Kenneth H. Rosen, Elementary number theory & its applications, sixth edition, Addison-Wesley, ISBN 978 0-321-50031-1
2. Ronald E. Walpole, Raymond H. Myers, Sharon L. Myers, Keying Ye, Probability & StatisticsforEngineers&Scientists,9th Ed. Pearson Publishers.
3. S. D. Sharma, Operations Research, Kedarnath and Ramnath Publishers, Meerut, Delhi

REFERENCE BOOK:

1. S C Gupta and V K Kapoor, Fundamentals of Mathematical statistics, Khanna publications
2. T.T. Soong, Fundamentals of Probability And Statistics For Engineers, John Wiley & Sons Ltd, 2004.
3. Sheldon M Ross, Probability and statistics for Engineers and scientists, Academic Press.

22CS2111: DATA STRUCTURES

B.Tech II Year I Sem.

L	T	P	C
3	0	0	3

Prerequisites: Programming for Problem Solving.

Course Objectives

1. Exploring basic data structures such as linked lists, stacks and queues.
2. Introduces about dictionaries and hash table representations.
3. Introduces about various search trees such as BST, B trees , B+ trees, AVL trees and Red Black trees.
4. To understand the concepts of graphs and sorting techniques.
5. Introduces about different pattern matching and tries algorithms.

Course Outcomes

1. Select the data structures that efficiently model the information in a problem and implement stacks, queues using arrays and linked list.
2. Summarize efficiency trade-offs among different data structure implementations or combinations and implement Dictionaries, Hash tables.
3. Identify and design advance data structure using Non linear data structure such as search trees.
4. Analyze appropriate sorting technique for given problem and also implement graph traversal methods.
5. Implement the concepts of Pattern Matching and Tries

UNIT - I

Introduction to Data Structures, abstract data types, Linear list – singly linked list implementation, insertion, deletion and searching operations on linear list, Stacks- Operations, array and linked representations of stacks, stack applications, Queues- operations, array and linked representations.

UNIT - II

Dictionaries: linear list representation, skip list representation, operations - insertion, deletion and searching.
Hash Table Representation: hash functions, collision resolution-separate chaining, open addressing- linear probing, quadratic probing, double hashing, rehashing, extendible hashing.

UNIT - III

Search Trees: Binary Search Trees, Definition, Implementation, Operations- Searching, Insertion and Deletion, B- Trees, B+ Trees, AVL Trees, Definition, Height of an AVL Tree, Operations – Insertion, Deletion and Searching, Red –Black, Splay Trees.

UNIT - IV

Graphs: Graph Implementation Methods. Graph Traversal Methods.
Sorting: Quick Sort, Heap Sort, External Sorting- Model for external sorting, Merge Sort.

UNIT - V

Pattern Matching and Tries: Pattern matching algorithms-Brute force, the Boyer –Moore algorithm, the Knuth-Morris-Pratt algorithm, Standard Tries, Compressed Tries, Suffix tries.

TEXT BOOKS:

1. Fundamentals of Data Structures in C, 2nd Edition, E. Horowitz, S. Sahni and Susan Anderson Freed, Universities Press.
2. Data Structures using C– A.S.Tanenbaum, Y.Langsam, and M.J.Augenstein, PHI / Pearson Education.

REFERENCE BOOK:

1. Data Structures: A Pseudo code Approach with C, 2nd Edition, R.F. Gilberg and B.A. Forouzan, Cengage Learning.

22CS2112: COMPUTER ORGANIZATION AND ARCHITECTURE

B.Tech II Year I Sem.

L	T	P	C
3	0	0	3

Pre-requisite: A Course on "Digital Electronics".

Course Objectives To learn

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

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

1. Able to explain the principles of computer organization and simple register transfer language to specify various computer operations.
2. Able to compare different addressing modes and instruction formats.
3. Able to make use of all computer arithmetic operations.
4. Able to decide the type of data representations and types of memory organizations.
5. 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

Microprogrammed 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, Inter processor arbitration, Inter processor communication and synchronization, Cache Coherence.

TEXT BOOK:

1. Computer System Architecture – M. Morris Mano, Third Edition, Pearson/PHI.

REFERENCE BOOKS:

1. Computer Organization – Carl Hamacher, Zvonks Vranesic, SafeaZaky, Vth Edition, McGraw Hill.
2. Computer Organization and Architecture – William Stallings Sixth Edition, Pearson/PHI.
3. Structured Computer Organization – Andrew S. Tanenbaum, 4 th Edition ,PHI/Pearson

22AM2111: SOFTWARE ENGINEERING

B.Tech. II Year I Sem. **LT PC**

3 0 0 3

Course Objectives

- To understand software process models such as waterfall and evolutionary models.
- To understand software requirements and SRS document.
- To understand different software architectural styles.
- To understand software testing approaches such as unit testing and integration testing.
- To understand software risks, RMMM Plan, Risk strategies.

Course Outcomes

- Students should be able to describe software engineering framework activities and identify appropriate software Process models for the development of software.
- Students should be able to understand and list functional and non functional requirement specifications of a software project and design relevant software system models from the available software requirements.
- Students should be able to identify and apply appropriate software architectures and patterns to carry out high level design of a system e.g. UML and be able to critically compare alternative choices.
- Students should be able to deliver quality software products by analyzing software testing strategies and product metrics over the entire system life cycle.
- Students should be able to evaluate risks involved in a project management and develop simple, reactive and proactive risk strategies for a quality software.

UNIT-I

Introduction to Software Engineering: The evolving role of software, changing nature of software, software myths. **A Generic view of process**: Software engineering- a layered technology, a process framework, the capability maturity model integration (CMMI). **Process models**: The waterfall model, Spiral model and Agile methodology

UNIT-II

Software Requirements: Functional and non-functional requirements, user requirements, system requirements ,interface specification, the software requirements document.

Requirements engineering process: Feasibility studies, requirements elicitation and analysis ,requirements validation ,requirements management.

UNIT-III

Design Engineering: Design process and design quality, design concepts, the design model. Creating an architectural design: software architecture, data design, architectural styles and patterns, architectural design, conceptual model of UML ,basic structural modeling, class diagrams, sequence diagrams, collaboration diagrams, use case diagrams, component diagrams.

UNIT-IV

Testing Strategies: A strategic approach to software testing, test strategies for conventional software, black-box and white-box testing, validation testing, system testing, the art of debugging.

Metrics for Process and Products: Software measurement, metrics for software quality.

UNIT-V

Risk management: Reactive Vs proactive risk strategies, software risks, risk identification, risk projection, risk refinement, RMMM. **Quality Management:** Quality concepts, software quality assurance, software reviews, formal technical reviews, statistical software quality assurance, software reliability, the ISO 9000 quality standards.

TEXTBOOKS:

1. Software Engineering, A practitioner's Approach- Roger S. Pressman, 6th edition, Mc GrawHill International Edition.
2. Software Engineering- Sommerville,7th edition, Pearson Education.

REFERENCEBOOKS:

1. The unified modeling language user guide Grady Booch, James Rambaugh, Ivar Jacobson, Pearson Education.
2. Software Engineering, an Engineering approach- James F. Peters, Witold Pedrycz, John Wiley.
3. Software Engineering principles and practice- Waman S Jawadekar, The McGraw- Hill Companies.
4. Fundamentals of object-oriented design using UML Meilerpage - Jones: Pearson Education.

22AM2112: OPERATING SYSTEMS

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

3 0 0 3

Prerequisites:

1. A course on "Computer Programming and Data Structures".
2. A course on "Computer Organization and Architecture".

Course Objectives:

- Introduce operating system concepts (i.e.,processes, threads, scheduling, synchronization, deadlocks, memory management, file and I/O sub systems and protection)
- Introduce the issues to be considered in the design and development of operating system
- Introduce basic Unix commands, system call interface for process management, interprocess communication and I/O in Unix
- To analyze memory management schemes.
- To understand I/O management and File systems.

Course Outcomes:

- Will be able to control access to a computer and the files that may be shared
- Analyze various scheduling algorithms.
- Demonstrate the knowledge of the components of computers and their respective roles in computing.
- Ability to recognize and resolve user problems with standard operating environments.
- Gain practical knowledge of how programming languages, operating systems, and architectures interact and how to use each effectively.

UNIT-I

Operating System - Introduction, Structures-Simple Batch, Multiprogrammed, Time-shared, Personal Computer, Parallel, Distributed Systems, Real-Time Systems, System components, Operating System services, System Calls

Process - Process concepts and scheduling, Operations on processes, Cooperating Processes, Threads

UNIT-II

CPU Scheduling – Scheduling Criteria, Scheduling Algorithms, Multiple – Processor Scheduling. System call interface for process management- fork, exit, wait, waitpid, exec

Deadlocks - System Model, Deadlocks Characterization, Methods for Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, and Recovery from Deadlock

UNIT-III

Process Management and Synchronization –The Critical Section Problem, Synchronization Hardware, Semaphores, and Classical Problems of Synchronization, Critical Regions, Monitors **Interprocess Communication Mechanisms:** IPC between processes on a single computer system, IPC between processes on different systems, using pipes, FIFOs, message queues, shared memory.

UNIT-IV

Memory Management and Virtual Memory - Logical versus Physical Address Space, Swapping, Contiguous Allocation, Paging, Segmentation, Segmentation with Paging, Demand Paging, Page Replacement, Page Replacement Algorithms.

UNIT-V

File System Interface and Operations – Access methods, Directory Structure, Protection, File System Structure, Allocation methods, Free-space Management. Usage of open, create, read, write, close, lseek, stat,ioctl system calls.

TEXT BOOKS:

1. Operating System Principles – Abraham Silberchatz, Peter B. Galvin, GregGagne7thEdition, John Wiley.
2. Advanced programming in the UNIX environment, W.R.Stevens, Pearson education.

REFERENCE BOOKS:

1. Operating Systems – Internals and Design Principles, William 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.

22AM2151: INTRODUCTION TO DATA STRUCTURES LAB

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

0 0 2 1

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.
- It provides an ability to the students to identify and apply the suitable data structure for the given real world problem.
- It enables them to gain knowledge in practical applications of data structures .

Course Outcomes:

- Ability to develop C programs for computing and real-life applications using basic elements like control
- · Be able to design and analyze the time and space efficiency of the data structure statements, arrays, functions, pointers and strings, and data structures like stacks, queues and linked lists.
- Ability to Implement searching and sorting algorithms.
- Be capable to identity the appropriate data structure for given problem
- Have practical knowledge on the applications of data structures.

List of Experiments:

1. Write a program that uses functions to perform the following operations on singly linked list.:
   i) Creation ii)Insertion iii)Deletion iv)Traversal
2. Write a program that uses functions to perform the following operations on doubly linked list.:
   i) Creation ii)Insertion iii)Deletion iv)Traversal
3. Write a program that uses functions to perform the following operations on circular linked list.:
   i) Creation ii)Insertion iii)Deletion iv)Traversal
4. Write a program that implement stack (its operations)using
   i) Arrays ii)Pointers
5. Write a program that implement Queue (its operations)using
   i) Arrays ii)Pointers
6. Write a program that implements the following sorting methods to sort a given list of integers in ascending order
   i) Quick sort ii) Heap sort iii)Merge sort
7. Write a program to implement the tree traversal methods (Recursive and Non Recursive).
8. Write a program to implement
   i) Binary Search tree ii) B Trees iii) B+ Trees iv) AVL trees v)Red –Black trees
9. Write a program to implement the graph traversal methods.
10. Implement a Pattern matching algorithms using Boyer-Moore, Knuth-Morris-Pratt

TEXTBOOKS:

1. Fundamentals of Data Structures in C, 2nd Edition, E. Horowitz,S. Sahni and Susan Anderson Freed, Universities Press.
2. Data Structures using C – A.S.Tanenbaum, Y. Langsam, and M.J. Augenstein, PHI/Pearson Education.

REFERENC EBOOK:

1.Data Structures: A Pseudo code Approach with C, 2nd Edition, R.F.Gilberg and B.A.Forouzan, Cengage Learning.

22AM2152: OPERATING SYSTEMS LAB

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

0 0 2 1

Prerequisites: A course on "Programming for Problem Solving", A course on "Computer Organization and Architecture".

Co-requisite: A course on "Operating Systems".

Course Objectives:

- To provide an understanding of the design aspects of operating system concepts through simulation
- Introduce basic Unix commands, system cal l interface for process management, inter process communication and I/O in Unix
- Student will learn various process and CPU scheduling Algorithms through simulation programs
- Student will have exposure to System calls and simulate them.
- Student will learn deadlocks and process management & Inter Process communication and simulate

Course Outcomes:

- 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.
- Ability to design and solve synchronization problems.
- Ability to implement inter process communication between two processes.
- Implement paging replacement and disk scheduling techniques.

List of Experiments:

1. Write C programs to simulate the following CPU Scheduling algorithms a) FCFS b) SJF c) Round Robin d) Priority

2. Write programs using the I/O system calls of UNIX/LINUX operating system (open, read, write, close, fcntl, seek, stat, open dir, read dir)

3. Write a C program to simulate Bankers Algorithm for Deadlock Avoidance and Prevention.

4. Write a C program to implement the Producer–Consumer problem using semaphores using UNIX/LINUX system calls.

5. Write C programs to illustrate the following IPC mechanisms a) Pipes b) FIFOs c)Message Queues d) Shared Memory

6. Write C programs to simulate the following memory management techniques a)Paging b)Segmentation

7. Write C programs to simulate Page replacement policies a)FCFS b) LRU c) Optimal

TEXTBOOKS:

1. Operating System Principles – Abraham Silberchatz, Peter B.Galvin, GregGagne7thEdition, John Wiley
2. Advanced programming in the Unix environment, W.R. Stevens, Pearson education.

REFERENCEBOOKS:

1. Operating Systems – Internals and Design Principles, William 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

22AM2153: SOFTWARE ENGINEERING LAB

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

0 0 2 1

Prerequisites

- A course on "Programming for Problem Solving".

Co-requisite

- A Course on "Software Engineering".

Course Objectives:

- To be able to write a problem statement for the given system.
- To have hands on experience in developing a software project by using various software engineering principles and methods in each of the phases of software development.
- To gain knowledge about source tools used for implementing software methods.
- To Specify Software Requirement Using Data Flow Diagram.
- Provide a scope to the students where they can solve small, real life problems

Course Outcomes:

- Develop the Problem Statement for the given system.
- Ability to translate end-user requirements into system and software requirements
- Capture the Requirements Specification for an intended Software System Using DFD.
- Ability to generate a high-level design of the system from the software requirements
- Will have experience and/or awareness of testing problems and will be able to develop a simple testing report

List of Experiments

Do the following seven exercises for any two projects given in the list of sample projects or any other Projects:

1. Development of problem statements.
2. Preparation of Software Requirement Specification Document, Design Documents and Testing Phase related documents.
3. Preparation of Software Configuration Management and Risk Management related documents.
4. Study and usage of any Design phase CASE tool
5. Performing the Design by using any Design phase CASE tools.
6. Develop test cases for unit testing and integration testing
7. Develop test cases for various white box and black box testing techniques.

Sample Projects:

1. Passport automation System
2. Book Bank
3. Online Exam Registration
4. Stock Maintenance System
5. Online course reservation system
6. E-ticketing
7. Software Personnel Management System
8. Credit Card Processing
9. E-book management System.
10. Recruitment system

TEXTBOOKS:

1. Software Engineering, A practitioner's Approach- Roger S. Pressman, 6th edition, McGraw Hill International Edition.
2. Software Engineering- Sommerville, 7th edition, Pearson Education.
3. The unified modeling language user guide Grady Booch, James Rambaugh, Ivar Jacobson, Pearson Education.

REFERENCE BOOKS:

1. Software Engineering, an Engineering approach- J ames F.Peters, Witold Pedrycz, John Wiley.
2. Software Engineering principles and practice- Waman S Jawadekar, The McGraw - Hill

22AM2154: NODEJS/ REACT JS/ DJANGO

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

0 0 2 1

Prerequisites: Object Oriented Programming through Java, HTML Basics

Course Objectives:

- To implement the static web pages using HTML and do client side validation using Java Script.
- To design and work with databases using Java
- To develop an end to end application using java full stack.
- To introduce Node JS implementation for server side programming.
- To experiment with single page application development using React.

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

- Build a custom website with HTML, CSS, and Boots trap and little JavaScript.
- Demonstrate Advanced features of Java Script and learn about JDBC
- Develop Server–side implementation using Java technologies like
- Develop the server–side implementation using Node JS.
- Design a Single Page Application using React.

Exercises:

1. Build a responsive web application for shopping cart with registration, login, catalog and cart pages using CSS3features, flex and grid.
2. Make the above web application responsive web application using Bootstrap framework.
3. Use JavaScript for doing client–side validation of the pages implemented in experiment 1 and experiment 2.
4. Explore the features of ES6 like arrow functions, callbacks, promises, as y nc/await. Implement an application for reading the weather information from openweathermap.org and display the information in the form of a graph on the web page.
5. Develop a java stand alone application that connects with the database (Oracle / my Sql) and perform the CRUD operation on the database tables.
6. Create an xml for the book store. Validate the same using both DTD and XSD.
7. Design a controller with servlet that provides the interaction with application developed inexperiment1and the database created in experiment 5.
8. Maintaining the transaction al history of any user is very important. Explore the various session tracking mechanism (Cookies, HTTP Session)
9. Create a custom server using http module and explore the other modules of Node JS like OS, path, event.
10. Develop an express web application that can interact with RESTAPI to perform CRUD operations on student data.(Use Postman)
11. For the above application create authorized end points using JWT (JSON Web Token).
12. Create a react application for the student management system having registration, login, contact, about pages and implement trouting to navigate through these pages.
13. Create a service in react that fetches the weather information from openweathermap.org and the display the current and historical weather information using graphical representation using chart.js
14. Create a TODO application in react with necessary components and deploy it into github.

REFERENCE BOOKS:

1. Jon Duckett, Beginning HTML, XHTML, CSS, and Java Script, Wrox Publications, 2010
2. Bryan Basham, Kathy Sierra and Bert Bates, Head First Servlets and JSP, O'Reilly Media, 2nd Edition, 2008.
3. Vasan Subramanian, Pro MERN Stack, Full Stack Web App Development with Mongo, Express, React, and Node, 2nd Edition, A Press.

22MC0003: CONSTITUTION OF INDIA

B.Tech. II Year I Sem. **LT P C**

3 0 0 0

Course Objectives: Students will be able to:

- Understand the premises informing the twin themes of liberty and freedom from a civil rights perspective.
- To address the growth of Indian opinion regarding modern Indian intellectuals' constitutional role and entitlement to civil and economic rights as well as the emergence of nationhood in the early years of Indian nationalism.
- To address the role of socialism in India after the commencement of the Bolshevik Revolution in 1917and its impact on the initial drafting of the Indian Constitution.

Course Outcomes: Students will be able to:

- Discuss the growth of the demand for civil rights in India for the bulk of Indians before the arrival of Gandhi in Indian politics.
- Discuss the intellectual origins of the framework of argument that informed the conceptualization of social reforms leading to revolution in India.
- Discuss the circumstances surrounding the foundation of the Congress Socialist Party [CSP] under the leadership of Jawaharlal Nehru and the eventual failure of the proposal of direct elections through adult suffrage in the Indian Constitution
- Discuss the passage of the Hindu Code Bill of 1956.

Unit-1 History of Making of the Indian Constitution- History of Drafting Committee.

Unit- 2 Philosophy of the Indian Constitution- Preamble Salient Features

Unit- 3 Contours of Constitutional Rights & Duties- Fundamental Rights

- Right to Equality
- Right to Freedom
- Right against Exploitation
- Right to Freedom of Religion
- Cultural and Educational Rights
- Right to Constitutional Remedies
- Directive Principles of State Policy
- Fundamental Duties.

Unit - 4 Organs of Governance: Parliament, Composition, Qualifications and Disqualifications, Powers and Functions, Executive, President, Governor, Council of Ministers, Judiciary, Appointment and Transfer of Judges, Qualifications, Powers and Functions

Unit - 5 Local Administration: District's Administration head: Role and Importance, Municipalities: Introduction, Mayor and role of Elected Representative, CEO of Municipal Corporation. Panchayat raj: Introduction, PRI: Zila Panchayat. Elected officials and their roles, CEO Zila Panchayat: Position and role. Block level: Organizational Hierarchy (Different departments), Village level: Role of Elected and Appointed officials, Importance of grass root democracy

Unit -6 Election Commission: Election Commission: Role and Functioning. Chief Election Commissioner and Election Commissioners. State Election Commission: Role and Functioning. Institute and Bodies for the welfare of SC/ST/OBC and women.

Suggested Reading:

1. The Constitution of India, 1950 (Bare Act),Government Publication.
2. Dr. S. N. Busi, Dr.B. R. Ambedkar framing of Indian Constitution, 1st Edition, 2015.
3. M. P. Jain, Indian Constitution Law, 7th Edn., Lexis Nexis, 2014.
4. D.D. Basu, Introduction to the Constitution of India, Lexis Nexis, 2015.

Counselling Code : VBIT

VIGNANA BHARATHI Institute of Technology®

(A UGC Autonomous Institution, Approved by AICTE, Accredited by NBA & NAAC-A Grade, Affiliated to JNTUH)

II YEAR – SEM-II

22BS2211: DISCRETE MATHEMATICS

B.Tech. II Year II Sem. **LT PC**

3 0 0 3

Course Objectives:

- Introduces elementary discrete mathematics for computer science and engineering.
- To introduce generating functions and recurrence relations.
- To perform the operations associated with sets, functions, and relations.
- Topics include formal logic notation, methods of proof, induction, sets, relations, algebraic structures, elementary graph theory, permutations and combinations, counting principles; recurrence relations and generating functions.
- To use Graph Theory for solving problems

Course Outcomes:

- Understand and construct precise mathematical proofs
- Apply logic and set theory to formulate precise statements
- Analyze and solve counting problems on finite and discrete structures
- Describe and manipulate sequences
- Apply graph theory in solving computing problems

UNIT-I

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

UNIT-II

Set theory: Introduction, Basic Concepts of Set Theory, Representation of Discrete Structures, Relations and Ordering, Functions.

UNIT-III

Algebraic Structures: Introduction, Algebraic Systems, Semi groups and Monoids, Lattices as Partially Ordered Sets, Boolean algebra.

UNIT-IV

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

UNIT-V

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

TEXTBOOKS:

1. Discrete Mathematical Structures with Applications to Computer Science: J.P. Tremblay, R.Manohar,McGraw-Hill,1st ed.
2. Discrete Mathematics for Computer Scientists & Mathematicians: Joel.Mott, Abraham Kandel, Teodore P. Baker, Prentis Hall ofIndia, 2nd ed.

REFERENCEBOOKS:

1. Discrete and Combinatorial Mathematics - an applied introduction: Ralph.P. Grimald, Pearson education ,5th edition.
2. Discrete Mathematical Structures: Thomas Kosy, Tata Mc GrawHill publishing co.

22AM2211: AUTOMATA THEORY AND COMPILER DESIGN

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

3 0 0 3

Course Objectives

- To introduce the fundamental concepts of formal languages, grammars and automata theory.
- To understand deterministic and non-deterministic machines and the differences between decidability and undecidability.
- Introduce the major concepts of language translation and compiler design and impart the knowledge of practical skills necessary for constructing a compiler.
- Topics include phases of compiler, parsing, syntax directed translation, type checking use of symbol tables, intermediate code generation.
- To provide the understanding of language translation peculiarities by designing complete translator for mini language.

Course Outcomes

- Able to employ finite state machines for modeling and solving computing problems.
- Able to design context free grammars for formal languages.
- Able to distinguish between decidability and undecidability.
- Demonstrate the knowledge of patterns, tokens & regular expressions for lexical analysis.
- Acquire skills in using lex tool and design LR parsers

UNIT-I

Introduction to Finite Automata: Structural Representations, Automata and Complexity, the Central Concepts of Automata Theory– Alphabets, Strings, Languages, Problems.

Nondeterministic Finite Automata: Formal Definition, an application, TextSearch, Finite Automata with Epsilon-Transitions.

Deterministic Finite Automata: Definition of DFA, How A DFA Process Strings, The language of DFA, Conversion of NFA with €-transitions to NFA without €-transitions. Conversion of NFA to DFA

UNIT-II

Regular Expressions: Finite Automata and Regular Expressions, Applications of Regular Expressions, Algebraic Laws for Regular Expressions, Conversion of Finite Automata to Regular Expressions.

Pumping Lemma for Regular Languages: Statement of the pumping lemma, Applications of the Pumping Lemma.

Context-Free Grammars: Definition of Context-Free Grammars, Derivations Using a Grammar ,Leftmost and Rightmost Derivations, the Language of a Grammar, Parse Trees, Ambiguity in Grammars and Languages.

UNIT-III

Push Down Automata: Definition of the Push down Automaton, the Languages of a PDA, Equivalence of PDA and CFG's, Acceptance by final state

Turing Machines: Introduction to Turing Machine, Formal Description, Instantaneous description, The language of a Turing machine

Undecidability: Undecidability, A Language that is Not Recursively Enumerable, An Undecidable Problem That is RE, Undecidable Problems about Turing Machines

UNIT-IV

Introduction: The structure of a compiler,

Lexical Analysis: The Role of the Lexical Analyzer, Input Buffering, Recognition of Tokens, The Lexical-Analyzer Generator Lex,

Syntax Analysis: Introduction, Context-Free Grammars, Writing a Grammar, Top-Down Parsing, Bottom-Up Parsing, Introduction to LR Parsing: Simple LR, More Powerful LR Parsers

UNIT-V

Syntax-Directed Translation: Syntax-Directed Definitions, Evaluation Orders for SDD's, Syntax-Directed Translation Schemes, Implementing L-Attributed SDD's.

Intermediate-Code Generation: Variants of Syntax Trees, Three- Address Code

Run-Time Environments: Stack Allocation of Space, Access to Nonlocal Data on the Stack, Heap Management

TEXT BOOKS:

1. Introduction to Automata Theory, Languages ,and Computation, 3nd Edition, John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman, Pearson Education.
2. Theory of Computer Science–Automata languages and computation, Mishra and Chandrashekaran, 2nd Edition, PHI.

REFERENCE BOOKS:

1. Compilers: Principles, Techniques and Tools, Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffry D.Ullman, 2nd Edition, Pearson.
2. Introduction to Formal languages Automata Theory and Computation ,Kamala Krithi vasan, Rama R, Pearson.
3. Introduction to Languages and The Theory of Computation, John C Martin, TMH.
4. Lex & yacc–John R. Levine, Tony Mason, Doug Brown, O'reilly Compiler Construction ,Kenneth C. Louden, Thomson. Course Technology.

VIGNANA BHARATHI Institute of Technology
(A UGC Autonomous Institution, Approved by AICTE, Accredited by NBA & NAAC-A Grade, Affiliated to JNTUH)

22CS2212: DATABASE MANAGEMENT SYSTEMS

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

3 0 0 3

Prerequisites: A course on "Data Structures".

Course Objectives:

- To understand the basic concepts and the applications of database systems.
- To recognize the importance of database analysis and design in the implementation of any Database application and to understand the process of drawing the ER-Diagrams.
- To master the basics of SQL and construct queries using SQL.
- Topics include data models, database design, relational model, relational algebra, transaction control, concurrency control, storage structures and access techniques.
- Familiar with database storage structures and access techniques

Course Outcomes:

- Gain knowledge of fundamentals of DBMS, database design and normal forms
- Master the basics of SQL for retrieval and management of data.
- Be acquainted with the basics of transaction processing and concurrency control.
- Familiarity with database storage structures and access techniques
- Improve the database design by normalization.

UNIT-I

Database System Applications: A Historical Perspective, File Systems versus a DBMS, the Data Model, Levels of Abstraction in a DBMS, Data Independence, Structure of a DBMS

Introduction to Database Design: Database Design and ER Diagrams, Entities, Attributes, and Entity Sets, Relationships and Relationship Sets, Additional Features of the ER Model, Conceptual Design With the ER Model

UNIT-II

Introduction to the Relational Model: Integrity constraint over relations, enforcing integrity constraints, querying relational data, logical database design, introduction to views, destroying /altering tables and views. Relational Algebra, Tuple relational Calculus, Domain relational calculus.

UNIT-III

SQL: QUERIES, CONSTRAINTS, TRIGGERS: form of basic SQL query, UNION, INTERSECT, and EXCEPT, Nested Queries, aggregation operators, NULL values, complex integrity constraints in SQL, triggers and active databases.

Schema Refinement: Problems caused by redundancy, decompositions, problems related to decomposition, reasoning about functional dependencies, First, Second, Third normal forms, BCNF, lossless join decomposition, multivalued dependencies, Fourth normal form, Fifth normal form.

UNIT-IV

Transaction Concept, Transaction State, Implementation of Atomicity and Durability, Concurrent Executions, Serializability, Recoverability, Implementation of Isolation, Testing for serializability, LockBased Protocols, Timestamp Based Protocols, Validation- Based Protocols, Multiple Granularity, Recovery and Atomicity,Log–Based Recovery, Recovery with Concurrent Transactions.

UNIT-V

Data on External Storage, File Organization and Indexing, Cluster Indexes, Primary and Secondary Indexes, Index data Structures, Hash Based Indexing, Tree based Indexing, Comparison of File Organizations, Indexes-Intuitions for tree Indexes, Indexed Sequential Access Methods(ISAM),
B+Trees: A Dynamic Index Structure.

TEXT BOOKS:

1. Database System Concepts, Silberschatz, Korth, Mc Grawhill, V edition. 3rd Edition
2. Database Management Systems, Raghurama Krishnan, Johannes Gehrke, Tata McGraw Hill

REFERENCE BOOKS:

1. Database Systems design, Implementation, and Management, Peter Rob & Carlos Coronel 7th Edition.
2. Fundamentals of Database Systems, Elmasri Navrate, Pearson Education
3. Introduction to Database Systems, C.J. Date, Pearson Education
4. Oracle for Professionals, The X Team, S.Shah and V. Shah, SPD.
5. Database Systems Using Oracle: A Simplified guide to SQL and PL/SQL, Shah, PHI.
6. Fundamentals of Database Management Systems, M.L. Gillenson, Wiley Student Edition.

VIGNANA BHARATHI Institute of Technology
(A UGC Autonomous Institution, Approved by AICTE, Accredited by NBA & NAAC-A Grade, Affiliated to JNTUH)

22AM2212: INTRODUCTION TO ARTIFICIAL INTELLIGENCE

B.Tech. II Year II Sem.

L	T	P	C
3	0	0	3

Prerequisite: Knowledge on Data Structures.

Course Objectives:

- To review and strengthen important mathematical concepts required for AI & ML.
- To learn the distinction between optimal reasoning Vs. human like reasoning.
- To understand the concepts of state space representation, exhaustive search, heuristic Search together with the time and space complexities.
- To learn different knowledge representation techniques.
- To understand the applications of AI, namely game playing, theorem proving and machine learning.

Course Outcomes:

- Learn the distinction between optimal reasoning Vs human like reasoning and formulate an efficient problem space for a problem expressed in natural language. Also select a search algorithm for a problem and estimate its time and space complexities.
- Apply AI techniques to solve problems of game playing, theorem proving and machine learning.
- Learn different knowledge representation techniques.
- Understand the concepts of state space representation, exhaustive search, heuristic search together with the time and space complexities.
- Comprehend the applications of Probabilistic Reasoning and Bayesian Networks.

UNIT-I

Introduction to AI-Intelligent Agents, Problem-Solving Agents,

Searching for Solutions - Breadth-first search, Depth-first search, Hill-climbing search, Simulated annealing search, Local Search in Continuous Spaces.

UNIT-II

Games-Optimal Decisions in Games, Alpha–Beta Pruning, Defining Constraint Satisfaction Problems, Constraint Propagation, Backtracking Search for CSPs, Knowledge-Based Agents, **Logic**-Propositional Logic, Propositional Theorem Proving: Inference and proofs, Proof by resolution, Horn clauses and definite clauses.

UNIT-III

First-Order Logic -Syntax and Semantics of First-Order Logic, Using First Order Logic, Knowledge Engineering in First-Order Logic. Inference in First-Order Logic: Propositional vs. First-Order Inference, Unification, Forward Chaining, Backward Chaining, Resolution.

Knowledge Representation: Ontological Engineering, Categories and Objects, Events.

UNIT-IV

Planning-Definition of Classical Planning ,Algorithms for Planning with State Space Search, Planning Graphs, other Classical Planning Approaches, Analysis of Planning approaches. Hierarchical Planning.

UNIT-V

Probabilistic Reasoning:

Acting under Uncertainty, Basic Probability Notation Bayes' Rule and Its Use, Probabilistic Reasoning , Representing Knowledge in an Uncertain Domain, The Semantics of Bayesian Networks, Efficient Representation of Conditional Distributions, Approximate Inference in Bayesian Networks, Relational and First-Order Probability.

TEXTBOOK:

1. Artificial Intelligence: A Modern Approach, Third Edition, Stuart Russell and Peter Norvig, Pearson Education.

REFERENCEBOOKS:

1. Artificial Intelligence,3rd Edn.,E. Richand K.Knight (TMH)
2. Artificial Intelligence,3rd Edn., Patrick Henny Winston ,Pearson Education.
3. Artificial Intelligence, Shivani Goel, Pearson Education.
4. Artificial Intelligence and Expert systems–Patterson, Pearson Education.

22AM2213: OBJECT ORIENTED PROGRAMMING THROUGH JAVA

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

3 0 0 3

Course Objectives

- To understand the basic object-oriented programming concepts and apply them in problem solving.
- To illustrate inheritance concepts for reusing the program.
- To Demonstrate multitasking by using multiple threads and event handling
- To develop data-centric applications using JDBC.
- To Understand the basics of java console and GUI based programming

Course Outcomes

- Demonstrate the behavior of programs involving the basic programming constructs like control structures, constructors, string handling and garbage collection.
- Demonstrate the implementation of inheritance (multilevel, hierarchical and multiple) by using extend and implement keywords
- Use multithreading concepts to develop interprocess communication.
- Understand the process of graphical user interface design and implementation using AWT or swings.
- Develop applets that interact abundantly with the client environment and deploy on the server.

UNIT-I

Object oriented thinking and Java Basics- Need for oop paradigm, summary of oop concepts, coping with complexity, abstraction mechanisms. A way of viewing world – Agents, responsibility, messages, methods, History of Java, Java buzzwords, data types, variables, scope and lifetime of variables, arrays, operators, expressions, control statements, type conversion and casting ,simple java program, concepts of classes, objects, constructors, methods, access control, this keyword, garbage collection, overloading methods and constructors, method binding, inheritance, overriding and exceptions, parameter passing, recursion, nested and inner classes, exploring string class.

UNIT-II

Inheritance, Packages and Interfaces– Hierarchical abstractions, Base class object, sub class, sub type, substitutability, forms of inheritance specialization, specification, construction, extension, limitation, combination, benefits of inheritance, costs of inheritance. Member access rules, super uses, using final with inheritance, polymorphism- method overriding, abstract classes, the Object class. Defining, Creating and Accessing a Package, Understanding CLASSPATH, importing packages, differences between classes and interfaces, defining an interface, implementing interface, applying interfaces, variables in interface and extending interfaces. Exploring java.io.

UNIT-III

Exception handling and Multi threading—Concepts of exception handling, benefits of exception handling, Termination or resumptive models, exception hierarchy, usage of try, catch, throw, throws and finally, built in exceptions, creating own exception subclasses. String handling, exploring java.util. Differences between multithreading and multitasking, thread life cycle, creating threads, thread priorities, synchronizing threads, inter thread communication, thread groups, daemon threads, Enumerations, auto boxing, annotations, generics.

UNIT-IV

Event Handling: Events, Event sources, Event classes, Event Listeners, Delegation event model, handling mouse and keyboard events, Adapter classes. The AWT class hierarchy, user interface components-labels ,button, canvas, scrollbars, text components, checkbox, checkbox groups, choices, lists panels–scroll pane, dialogs , menu bar, graphics, layout manager–layout manager types–border, grid flow, card and grid bag.

UNIT-V

Applets – Concepts of Applets, differences between applets and applications, life cycle of an applet, types of applets, creating applets, passing parameters to applets. Swing – Introduction, limitations of AWT , MVC architecture, components, containers, exploring swing- JApplet, JFrame and JComponent, Icons and Labels, text fields, buttons– The JButton class, Checkboxes, Radio buttons, Combo boxes, Tabbed Panes, Scroll Panes, Trees and Tables.

TEXTBOOKS:

1. Java the complete reference, 7th edition, Herbert schildt, TMH.
2. Understanding OOP with Java, updated edition, T.Budd, Pearson education.

REFERENCEBOOKS:

1. An Introduction to programming and OO design using Java, J.Nino and F.A.Hosch, Johnwiley& sons.
2. An Introduction to OOP third edition, T. Budd, Pearson education.
3. Introduction to Java programming, Y.Daniel Liang, Pearson education.
4. An introduction to Java programming and object-oriented application development, R.A.Johnson-Thomson.
5. Core Java 2, Vol1, Fundamentals, Cay.S.Horstmann and Gary Cornell, eighth Edition, Pearson Education.
6. Core Java 2,Vol 2, Advanced Features, Cay.S.Horstmann and Gary Cornell, eighth Edition, Pearson Education
7. Object Oriented Programming with Java,R.Budyya,S.T.Selvi,X.Chu,TMH.
8. Java and Object Orientation, an introduction, John Hunt, second edition, Springer.9. Maurach's Beginning Java 2 JDK5, SPD.

VIGNANA BHARATHI Institute of Technology ®

(A UGC Autonomous Institution, Approved by AICTE, Accredited by NBA & NAAC-A Grade, Affiliated to JNTUH)

22CS2252: DATABASE MANAGEMENT SYSTEMS LAB

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

0 0 2 1

Co-requisites:"Database Management Systems"

Course Objectives:

- Introduce ER data model, database design and normalization
- Learn SQL basics for data definition and data manipulation
- Design database schema for a given application and apply normalization
- Acquire skills in using SQL commands for data definition and data manipulation.
- Develop solutions for database applications using procedures, cursors and triggers.

Course Outcomes:

- Design database schema for a given application and apply normalization
- Acquire skills using SQL commands for data definition and data manipulation.
- Apply the normalization techniques for development of application software to realistic problems
- Develop solutions for database applications using procedures, cursors and triggers
- Applying PL/SQL for processing database

List of Experiments:

1. Concept design with E-R Model
2. Relational Model
3. Normalization
4. Practicing DDL commands
5. Practicing DML commands
6. A. Querying (using ANY, ALL, UNION, INTERSECT, JOIN, Constraints etc.)
   B. Nested, Correlated sub-queries
7. Queries using Aggregate functions, GROUPBY, HAVING and Creation and dropping of Views.
8. Triggers (Creation of insert trigger, delete trigger, update trigger)
9. Procedures
10. Usage of Cursors

TEXTBOOKS:

1. Database Management Systems, Raghurama Krishnan, Johannes Gehrke, Tata McGrawHill,3rd Edition
2. Database System Concepts, Silberschatz, Korth, McGrawHill, V edition.

REFERENCEBOOKS:

1. Database Systems design, Implementation and Management, Peter Rob & Carlos Coronel 7th Edition.
2. Fundamentals of Database Systems, Elma sri Navrate, Pearson Education
3. Introduction to Database Systems, C.J. Date, Pearson Education
4. Oracle for Professionals, The X Team, S. Shah and V. Shah, SPD.
5. Database Systems Using Oracle: A Simplified guide to SQL and PL/SQL, Shah, PHI.
6. Fundamentals of Database Management Systems, M.L. Gillenson, Wiley Student Edition.

22AM2251: JAVA PROGRAMMING LAB

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

0 0 2 1

Course Objectives:

- To understand OOP principles.
- To understand the Exception Handling mechanism.
- To understand Java collection framework.
- To understand multithreaded programming.
- To understand swing controls in Java.

Course Outcomes:

- Able to write the programs for solving real world problems using Java OOP principles.
- Able to write programs using Exceptional Handling approach.
- : Create Packages and build applications using default packages.
- Able to write multithreaded applications.
- Able to write GUI programs using swing controls in Java.

List of Experiments:

1. Use Eclipse or Net bean platform and acquaint yourself 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 debugging step by step with a small program of about 10 to 15 lines which contains atleast one if else condition and a for loop.

2. Write a Java program to demonstrate the OOP principles.[i.e., Encapsulation, Inheritance, Polymorphism and Abstraction]

3. Write a Java program to handle checked and unchecked exceptions. Also, demonstrate the usage of custom exceptions in real time scenario.

4. Write a Java program on Random Access File class to perform different read and write operations.

5. WriteaJavaprogramtodemonstratetheworkingofdifferentcollectionclasses. [Use package structure to store multiple classes].

6. Write a program to synchronize the threads acting on the same object. [Consider the example of any reservations like railway, bus, movie ticket booking, etc.]

7. Write a program to perform CRUD operations on the student table in a database using JDBC.

8. 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 divided by zero.

9. 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]

REFERENCE BOOKS:

1. Java for Programmers, P.J.Deitel and H.M.Deitel, 10th Edition Pearson education.
2. Thinking in Java, Bruce Eckel, Pearson Education.
3. Java Programming, D. S. Malik and P.S. Nair, Cengage Learning.
4. Core Java, Volume 1,9th edition, Cay S. Horstmann and G Cornell, Pearson.

VIGNANA BHARATHI Institute of Technology
(A UGC Autonomous Institution, Approved by AICTE, Accredited by NBA & NAAC-A Grade, Affiliated to JNTUH)

22AM2252: PROLOG/LISP/PYSWIP

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

0 0 2 1

List of Programs:

1. Write simple fact for following:
   A. Ram likes mango.
   B. Seema is a girl.
   C. Bill likes Cindy.
   D. Rose is red.
   E. John owns gold
2. Write predicates one converts centigrade temperatures to Fahrenheit, the other checks if a temperature is below freezing.
3. Write a program to solve the Monkey Banana problem
4. WAP in turbo prolog for medical diagnosis and show the advantages and disadvantages of green and red cuts.
5. Write a program to solve the 4-Queen problem.
6. Write a program to solve traveling salesman problems.
7. Write a program to solve waterjug problems using Prolog.
8. Write simple Prolog functions such as the following. Take into account lists which are too short.
   --remove the Nth item from the list.—insert as the Nth item.
9. Assume the prolog predicate gt(A, B) is true when A is greater than B. Use this predicate to define the predicate addLeaf (Tree, X, NewTree) which is true if NewTree is the Tree produced by adding the item X in a leaf node. Tree and NewTree are binary search trees. The empty tree is represented by the item nil.
10. Write a Prolog predicate, count Lists(Alist, Ne, Nl), using accumulators, that is true when Nl is the number of items that are listed at the top level of Alist and Ne is the number of empty lists. Suggestion: First try to count the lists, or empty lists, then modify by adding the other counter.
11. Define a predicate memCount (AList, Blist, Count) that is true if Alist occurs Count times within Blist. Define without using an accumulator. Use "not" as defined in utilities.pro, to make similar cases are unique, or else you may get more than one count as an answer.

    Examples:
    memCount(a,[b,a],N).N
    =1;
    nomemCount(a,[b,[a,a,[a],c],a],N).
    N =4;
    nomemCount([a],[b,[a,a,[a],c],a],N).
    N =1;
    No

REFERENCEBOOK:

1. Programming for Artificial Intelligence, 3e, by BRATKO, WILEY

22MC0004: GENDER SENSITIZATION LAB

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

0 0 2 0

COURSE DESCRIPTION

This course offers an introduction to Gender Studies, an interdisciplinary field that asks critical questions about the meanings of sex and gender in society. The primary goal of this course is to familiarize students with key issues, questions and debates in Gender Studies, both historical and contemporary. It draws on multiple disciplines–such as literature, history, economics, psychology, sociology, philosophy, political science, anthropology and media studies–to examine cultural assumptions about sex, gender and sexuality.

This course integrates analysis of current events through student presentations, aiming to increase awareness of contemporary and historical experiences of women and of the multiple ways that sex and gender interact with race, class, caste, nationality and other social identities. This course also seeks to build an understanding and initiate and strengthen programmes combating gender-based violence and discrimination. The course also features several exercises and reflective activities designed to examine the concepts of gender, gender-based violence, sexuality, and rights. It will further explore the impact of gender-based violence on education, health and development.

Objectives of the Course

- 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 debate 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.

Learning 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 labor 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 Labor-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.

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 from engineering departments.**

- ***Classes will consist of a combination of activities: dialogue-based lectures, discussions, collaborative learning activities, group work and in-class assignments. Apart from the above prescribed book, Teachers can make use of any authentic materials related to the topics given in the syllabus on "Gender".***

ESSENTIAL READING: The Textbook, "*Towards a World of Equals: A Bilingual Textbook on Gender"* written by A. Suneetha, Uma Bhrugubanda, DuggiralaVasanta, Rama Melkote, Vasudha Nagaraj, Asma Rasheed, Gogu Shyamala, Deepa Sreenivas and SusieTharu **published by Telugu Akademi, Telangana Government in 2015.**

ASSESSMENT AND GRADING:

- Discussion & Classroom Participation: 20%
- Project/Assignment: 30%
- End Term Exam:50%