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, Inter processor 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
4. Data Structures using C, R.Thareja, Oxford University Press.

19DS2112 - PYTHON PROGRAMMING

B.Tech. II Year I Semester

L	T	P	C
2	1	-	2

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

Course Objectives:

- Learn Syntax and Semantics and create Functions in Python.
- Handle Strings and Files in Python.
- Understand Lists, Dictionaries and Regular expressions in Python.
- Implement Object Oriented Programming concepts in Python.
- Build Web Services and introduction to Network and Database Programming in Python.

Course Outcomes:

- Examine Python syntax and semantics and be fluent in the use of Python flow control and functions.
- Demonstrate proficiency in handling Strings and File Systems.
- Create, run and manipulate Python Programs using core data structures like Lists, dictionaries and use Regular Expressions.
- Interpret the concepts of Object-Oriented Programming as used in Python.
- Implement exemplary applications related to Network Programming, Web Services and databases in Python.

UNIT - I

Python Basics, Objects- Python Objects, Standard Types, Other Built-in Types, Internal Types, Standard Type Operators, Standard Type Built-in Functions, Categorizing the Standard Types, Unsupported Types ,Numbers - Introduction to Numbers, Integers, Floating Point Real Numbers, Complex Numbers, Operators, Built-in Functions, Related Modules Sequences - Strings, Lists, and Tuples, Mapping and Set Types

UNIT - II

FILES: File Objects, File Built-in Function [open()], File Built-in Methods, File Built-in Attributes, Standard Files, Command-line Arguments, File System, File Execution, Persistent Storage Modules, Related Modules

Exceptions: Exceptions in Python, Detecting and Handling Exceptions, Context Management,*Exceptions as Strings, Raising Exceptions, Assertions, Standard Exceptions, *Creating Exceptions, Why Exceptions (Now)?, Why Exceptions at All?, Exceptions and the sys Module, Related Modules

Modules: Modules and Files, Namespaces, Importing Modules, Importing Module Attributes, Module Built-in Functions, Packages, Other Features of Modules

UNIT - III

Regular Expressions: Introduction, Special Symbols and Characters, Res and Python Multithreaded Programming: Introduction, Threads and Processes, Python, Threads, and the Global Interpreter Lock, Thread Module, Threading Module, Related Modules

UNIT - IV

GUI Programming: Introduction, Tkinter and Python Programming, Brief Tour of Other GUIs, Related Modules and Other GUIs

WEB Programming: Introduction, Wed Surfing with Python, Creating Simple Web Clients, Advanced Web Clients, CGI-Helping Servers Process Client Data, Building CGI Application Advanced CGI, Web (HTTP) Servers

UNIT - V

Database Programming: Introduction, Python Database Application Programmer's Interface (DB-API),Object Relational Managers (ORMs), Related Modules

TEXT BOOK:

1. Core Python Programming, Wesley J. Chun, Second Edition, Pearson.

REFERENCE BOOKS:

1. Think Python, Allen Downey, Green Tea Press
2. Introduction to Python, Kenneth A. Lambert, Cengage
3. Python Programming: A Modern Approach, Vamsi Kurama, Pearson
4. Learning Python, Mark Lutz, O'Really. Python Programming

19AM2114: DATA STRUCTURES USING C

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

3 - - 3

Prerequisites:

- A course on "Programming for Problem Solving".
- C Language

Course Objectives

- Applying notations to analyse the performance of algorithms.
- Understand the concepts of linear and non-linear data structures Abstract Data Type (ADT)
- Implement the linear data structures such as stacks, queues.
- Implement the non-linear data structures such as trees and graphs.
- Analyse various searching and sorting techniques

Course Outcomes

- Able to analyse the time and space complexity of algorithms.
- Able to describe and summarize linear data structures ADT such as stacks and queues.
- Able to design and implement non-linear data structures ADT such as Trees and Graphs.
- Able to demonstrate and implement various kinds of searching and sorting techniques.
- Able to construct spanning trees.

UNIT- I

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

Linked Lists: Singly Linked Lists-Operations-Insertion, Deletion, Concatenating singly linked lists, Reversing singly Linked Lists, Circularly linked lists- Ooperations for Circularly linked lists, Doubly Linked Lists- Operations- Insertion, Deletion.

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 – Terminology, Sequential and linked Representations of Graphs: Adjacency Matrices, Adjacency List, Adjacency Multi list, Graph Traversal: Depth First Search and Breadth First Search.

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, Merge 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.

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.

REFERENCES:

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.

19MB2114 Business Economics and Financial Analysis

B.Tech. II Year I Semester

L	T	P	C
3	-	-	3

Course Objective:

- To learn the basic Business types, impact of the Economy on Business and Firms specifically.
- To analyze the Business from the Financial Perspective.

Course Outcome:

- The students will understand the various Forms of Business and the impact of economic variables on the Business. The Demand, Supply, Production, Cost,Market Structure, Pricing aspects are learnt.
- The Students can study the firm‟s financial position by analysing the Financial Statements of a Company.

UNIT – I

Introduction to Business and Economics:Business: Structure of Business Firm, Theory of Firm, Types of Business Entities, Limited Liability Companies, Sources of Capital for a Company, Non-Conventional Sources of Finance.

Economics: Significance of Economics, Micro and Macro Economic Concepts, Concepts and Importance of National Income, Inflation, Money Supply in Inflation, Business Cycle, Features and Phases of Business Cycle. Nature and Scope of Business Economics, Role of Business Economist, Multidisciplinary nature of Business Economics.

UNIT – II

Demand and Supply Analysis:Elasticity of Demand: Elasticity, Types of Elasticity, Law of Demand, Measurement and Significance of Elasticity of Demand, Factors affecting Elasticity of Demand, Elasticity of Demand in decision making, Demand Forecasting: Characteristics of Good Demand Forecasting, Steps in Demand Forecasting, Methods of Demand Forecasting.

Supply Analysis: Determinants of Supply, Supply Function & Law of Supply.

UNIT - III

Production, Cost, Market Structures & Pricing:Production Analysis: Factors of Production, Production Function, Production Function with one variable input, two variable inputs, Returns to Scale, Different Types of Production Functions.

Cost analysis: Types of Costs, Short run and Long run Cost Functions.

Market Structures: Nature of Competition, Features of Perfect competition, Monopoly, Oligopoly, and Monopolistic Competition.

Pricing: Types of Pricing, Product Life Cycle based Pricing, Break Even Analysis, and Cost Volume Profit Analysis.

UNIT - IV

Financial Accounting: Accounting concepts and Conventions, Accounting Equation, Double-Entry system of Accounting, Rules for maintaining Books of Accounts, Journal, Posting to Ledger, Preparation of Trial Balance, Elements of Financial Statements, Preparation of Final Accounts.

UNIT -V

Financial Analysis through Ratios: Concept of Ratio Analysis, Liquidity Ratios, Turnover Ratios, Profitability Ratios, Proprietary Ratios, Solvency, Leverage Ratios (simple problems). Introduction to Fund Flow and Cash Flow Analysis (simple problems).

Text Books:

1. D. D. Chaturvedi, S. L. Gupta, Business Economics - Theory and Applications, International Book House Pvt. Ltd. 2013.

2. Dhanesh K Khatri, Financial Accounting, Tata McGraw Hill, 2011.

3. Geethika Ghosh, Piyali Gosh, Purba Roy Choudhury, Managerial Economics, 2e, Tata McGraw Hill Education Pvt. Ltd. 2012.

Reference Books:

1. Paresh Shah, Financial Accounting for Management 2e, Oxford Press, 2015.

2. S. N. Maheshwari, Sunil K Maheshwari, Sharad K Maheshwari, Financial Accounting, 5e, Vikas Publications, 2013.

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.

19BS2116 - MATHEMATICAL AND STATISTICAL FOUNDATIONS

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

3 - - 3

Prerequisites: Mathematics courses of first year of study.

Course Objectives:

- The Number Theory basic concepts useful for cryptography etc
- 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 course, the student must be able to

- Apply the number theory concepts to cryptography domain
- 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 Normal Curve, 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, FDistribution.

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 & Statistics for Engineers & Scientists, 9th Ed. Pearson Publishers.
3. S. D. Sharma, Operations Research, Kedarnath and Ramnath Publishers, Meerut, Delhi

REFERENCE BOOKS:

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

19AM2151: DATA STRUCTURES USING C LAB

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

- - 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 Single Linked List:

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

Week 2: Write a program that uses functions to perform the following operations on double 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 to implement Binary Search Tree insertion and display.

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, andM.J.Augenstein, PHI/Pearson Education.

REFERENCES:

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

19DS2152 - PYTHON PROGRAMMING 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:

- To be able to introduce core programming basics and program design with functions using Python programming language.
- To understand a range of Object-Oriented Programming, as well as in-depth data and information processing techniques.
- To understand the high-performance programs designed to strengthen the practical expertise.

Course Outcomes:

- Student should be able to understand the basic concepts scripting and the contributions of scripting language
- Ability to explore python especially the object-oriented concepts, and the built in objects of Python.
- Ability to create practical and contemporary applications such as TCP/IP network programming, Web applications, discrete event simulations

List of Experiments:

1. Write a program to demonstrate different number data types in Python.
2. Write a program to perform different Arithmetic Operations on numbers in Python.
3. Write a program to create, concatenate and print a string and accessing sub-string from a given string.
4. Write a python script to print the current date in the following format "Sun May 29 02:26:23 IST 2017"
5. Write a program to create, append, and remove lists in python.
6. Write a program to demonstrate working with tuples in python.
7. Write a program to demonstrate working with dictionaries in python.
8. Write a python program to find largest of three numbers.
9. Write a Python program to convert temperatures to and from Celsius, Fahrenheit. [Formula : c/5 = f-32/9]
10. Write a Python program to construct the following pattern, using a nested for loop

```
*
* *
* * *
* * * *
* * * * *
* * * *
* * *
```

* *
*

11. Write a Python script that prints prime numbers less than 20.
12. Write a python program to find factorial of a number using Recursion.
13. Write a program that accepts the lengths of three sides of a triangle as inputs. The program output should indicate whether or not the triangle is a right triangle (Recall from the Pythagorean Theorem that in a right triangle, the square of one side equals the sum of the squares of the other two sides).
14. Write a python program to define a module to find Fibonacci Numbers and import the module to another program.
15. Write a python program to define a module and import a specific function in that module to another program.
16. Write a script named copyfile.py. This script should prompt the user for the names of two text files. The contents of the first file should be input and written to the second file.
17. Write a program that inputs a text file. The program should print all of the unique words in the file in alphabetical order.
18. Write a Python class to convert an integer to a roman numeral.
19. Write a Python class to implement pow(x, n)
20. Write a Python class to reverse a string word by word.

19MC0006: UNIVERSAL HUMAN VALUES - II

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

2 - - 0

Pre-requisites:

- None. Universal Human Values 1 (desirable)

Course Objectives

The objective of the course is four fold:

- Development of a holistic perspective based on self-exploration about
- Themselves (human being), family, society and nature/existence.
- Understanding (or developing clarity) of the harmony in the human being,
- family, society and nature/existence
- Strengthening of self-reflection.
- Development of commitment and courage to act.

COURSE TOPICS:

The course has 28 lectures and 14 practice sessions in 5 modules:

Module 1: Course Introduction - Need, Basic Guidelines, Content and Process for Value Education

1. Purpose and motivation for the course, recapitulation from Universal Human Values-I
2. Self-Exploration–what is it? - Its content and process; 'Natural Acceptance and Experiential Validation- as the process for self-exploration
3. Continuous Happiness and Prosperity- A look at basic Human Aspirations
4. Right understanding, Relationship and Physical Facility- the basic requirements for fulfilment of aspirations of every human being with their correct priority
5. Understanding Happiness and Prosperity correctly- A critical appraisal of the current scenario
6. Method to fulfill the above human aspirations: understanding and living inharmony at various levels.
   Include practice sessions to discuss natural acceptance in human being as theinnate acceptance for living with responsibility (living in relationship, harmony and co-existence) rather than as arbitrariness in choice based on liking-disliking

Module 2: Understanding Harmony in the Human Being - Harmony in Myself!

7. Understanding human being as a co-existence of the sentient 'I' and the material 'Body'
8. Understanding the needs of Self ('I') and 'Body' - happiness and physical facility
9. Understanding the Body as an instrument of 'I' (I being the doer, seer and enjoyer)
10. Understanding the characteristics and activities of 'I' and harmony in 'I'
11. Understanding the harmony of I with the Body: Sanyam and Health; correct appraisal of

Physical needs, meaning of Prosperity in detail

12. Programs to ensure Sanyam and Health.
    Include practice sessions to discuss the role others have played in making material goods available to me. Identifying from one's own life. Differentiate between prosperity and accumulation. Discuss program for ensuring health vs dealing with disease

Module 3: Understanding Harmony in the Family and Society- Harmony in Human-Human Relationship

13. Understanding values in human-human relationship; meaning of Justice (nine universal values in relationships) and program for its fulfilment to ensure mutual happiness; Trust and Respect as the foundational values of relationship
14. Understanding the meaning of Trust; Difference between intention and competence
15. Understanding the meaning of Respect, Difference between respect and differentiation; the other salient values in relationship
16. Understanding the harmony in the society (society being an extension of family):Resolution, Prosperity, fearlessness (trust) and co-existence as comprehensive Human Goals
17. Visualizing a universal harmonious order in society- Undivided Society, Universal Order- from family to world family.
    Include practice sessions to reflect on relationships in family, hostel and institute as extended family, real life examples, teacher-student relationship, goal of education etc. Gratitude as a universal value in relationships. Discuss with scenarios. Elicit examples from students' lives

Module 4: Understanding Harmony in the Nature and Existence - Whole existence as Coexistence

18. Understanding the harmony in the Nature
19. Interconnectedness and mutual fulfillment among the four orders of nature recycle ability and self-regulation in nature
20. Understanding Existence as Co-existence of mutually interacting units in all-pervasive space
21. Holistic perception of harmony at all levels of existence. Include practice sessions to discuss human being as cause of imbalance in nature (film "Home" can be used), pollution, depletion of resources and role of technology etc.

Module 5: Implications of the above Holistic Understanding of Harmony on Professional Ethics

22. Natural acceptance of human values
23. Definitiveness of Ethical Human Conduct
24. Basis for Humanistic Education, Humanistic Constitution and Humanistic Universal Order.
25. Competence in professional ethics:
    a. Ability to utilize the professional competence for augmenting universal human order

b. Ability to identify the scope and characteristics of people friendly and eco-friendly production systems,

c. Ability to identify and develop appropriate technologies and management patterns for above production systems.

26. Case studies of typical holistic technologies, management models and production systems
27. Strategy for transition from the present state to Universal Human Order:

a. At the level of individual: as socially and ecologically responsible engineers, technologists and managers

b. At the level of society: as mutually enriching institutions and organizations

28. Sum up.

Include practice Exercises and Case Studies will be taken up in Practice (tutorial)Sessions eg. To discuss the conduct as an engineer or scientist etc.

TEXT BOOKS:

1. Human Values and Professional Ethics by R R Gaur, R Sangal, G P Bagaria,Excel Books,

New Delhi, 2010

REFERENCES:

1. Jeevan Vidya: EkParichaya, A Nagaraj, Jeevan VidyaPrakashan, Amarkantak,1999.

2. Human Values, A.N. Tripathi, New Age Intl. Publishers, New Delhi, 2004.

3. The Story of Stuff (Book).

4. The Story of My Experiments with Truth - by Mohandas Karamchand Gandhi

5. Small is Beautiful - E. F Schumacher.

6. Slow is Beautiful - Cecile Andrews

7. Economy of Permanence - J C Kumarappa

8. Bharat Mein Angreji Raj - PanditSunderlal

9. Rediscovering India - by Dharampal

10. Hind Swaraj or Indian Home Rule - by Mohandas K. Gandhi

11. India Wins Freedom - Maulana Abdul Kalam Azad

12. Vivekananda - Romain Rolland (English)

13. Gandhi - Romain Rolland (English)

MODE OF CONDUCT

- Lectures hours are to be used for interactive discussion, placing the proposals about the topics at hand and motivating students to reflect, explore and verify them.
- Tutorial hours are to be used for practice sessions. While analyzing and discussing the topic, the faculty mentor's role is in pointing to essential elements to help in sorting them out from the surface elements. In other words, help the students explore the important or critical elements.
- In the discussions, particularly during practice sessions (tutorials), the mentor encourages the student to connect with one's own self and do self-observation, self-reflection and self-exploration.
- Scenarios may be used to initiate discussion. The student is encouraged to takeup "ordinary" situations rather than" extra-ordinary" situations. Such observations and their analyses are shared and discussed with other students and faculty mentor, in a group sitting.
- Tutorials (experiments or practical) are important for the course. The difference is that the laboratory is everyday life, and practical are how you behave and working real life. Depending on the nature of topics, worksheets, home assignment and/or activity are included. The practice sessions (tutorials) would also provide support to a student in performing actions commensurate to his/her beliefs. It is intended that this would lead to development of commitment, namely behaving and working based on basic human values.
- It is recommended that this content be placed before the student as it is, in theform of a basic foundation course, without including anything else or excludingany part of this content. Additional content may be offered in separate, highercourses.
- This course is to be taught by faculty from every teaching department,including HSS faculty.
- Teacher preparation with a minimum exposure to at least one 8-day FDP onUniversal Human Values is deemed essential.

ASSESSMENT:

- The assessment is to provide a fair state of development of the student, so participation in classroom discussions, self-assessment, peer assessment etc. will be used in evaluation.

Example:

Assessment by faculty mentor : 10 marks

Self-assessment : 10 marks

Assessment by peers : 10 marks

Socially relevant project/Group Activities/Assignments: 20 marks

The overall pass percentage is 40%. In case the student fails, he/she must repeat the course.

OUTCOME OF THE COURSE:

- By the end of the course, students are expected to become more aware of themselves, and their surroundings (family, society, nature); they would become more responsible in life, and in handling problems with sustainable solutions, while keeping human relationships and human nature in mind.
- They would have better critical ability. They would also become sensitive to their commitment towards what they have understood (human values, human relationship and human society). It is hoped that they would be able to apply what they have learnt to their own self in different day-to-day settings in real life, at least a beginning would be made in this direction.
- This is only an introductory foundational input. It would be desirable to follow it up by faculty-student or mentor-mentee programs throughout their time with the institution. Higher level courses on human values in every aspect of living. E.g. as a professional

IInd YEAR – SEM II

19CS2211 - OPERATING SYSTEMS

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

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

19CS2212- DATABASE MANAGEMENT SYSTEMS

B.Tech. II Year II Semester **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.

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.

19DS2214 - OBJECT ORIENTED PROGRAMMING THROUGH JAVA

B.Tech II Year II Semester

L	T	P	C
3	0	0	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, Array List, 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: Label, 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

REFERENCES:

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.

19AM2215 Software Engineering

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

3 - - 3

Prerequisites:

- Computer Programming
- Database Management Systems

Course Objectives

- To understand fundamental principles of Software engineering, and their application in the development of software products.
- To understand and create the software requirements specifications document.
- To understand and use unified modeling language for specifying, analysis and designing.
- To understand testing strategies for testing software applications
- To understand Software metrics and Risk Management strategies to identify potential problems before they occur.

Course Outcomes

- Able to apply the software engineering lifecycle phases communication, planning, analysis, design, construction, and deployment.
- Ability to translate end-user requirements into system and software requirements into Software Requirements specification Document (SRS)
- Able to apply UML in object-oriented software modeling to develop computer software.
- Able to identify problems in software and will be able to develop a simple testing report.
- To understand Software Metrics, potential risk and how to manage them though RMMM plan.

UNIT - I

Introduction: The evolving role of software, Changing Nature of Software, legacy software, Software myths.

A Generic view of process: Software engineering- A layered technology, a process framework, The Capability Maturity Model Integration (CMMI), Process patterns, process assessment, personal and team process models, COCOMO Model.

Process Models: The waterfall model, Incremental process models, Evolutionary process models, specialized process models, The Unified process.

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, Software requirements documents

System models: Context Models, Behavioral models, Data models, Object models, structured methods.

UNIT - III

Modeling Techniques using UML: The Unified Approach to Modeling, Structural and Behavioral Diagrams.

Design Engineering: Data Flow Diagrams, Design process and Design quality, Design concepts, the design model, pattern-based software design.

Creating an architectural design: Architectural styles and patterns, Architectural Design, assessing alternative architectural designs.

UNIT - IV

Implementation: Structured coding Techniques, Coding Styles-Standards and Guidelines.

Testing Strategies: A strategic approach to software testing, test strategies for conventional software, Regression Testing, Unit Testing, Black-Box and White-Box testing, Validation testing, System testing, the art of Debugging.

Quality Management: Quality concepts, software quality assurance, software reviews, statistical software quality assurance, software reliability, the ISO 9000 quality standards.

UNIT - V

Product metrics: Software quality, metrics for analysis model, metrics for design model, metrics for source code, metrics for testing, metrics for maintenance.

Risk management: Reactive Vs proactive risk strategies, software risks, risk identification, risk projection, risk refinement, RMMM, RMMM plan.

TEXT BOOKS:

1. Software Engineering, A practitioner's Approach- Roger S. Pressman, 6th edition, Mc Graw 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.

REFERENCES:

1. "Software Engineering", Ian Sommerville, Addison-Wesley, 9th Edition, 2010, ISBN- 13: 978- 0137035151.
2. Richard Fairley, "Software Engineering Concepts", Tata McGraw Hill.
3. Fundamentals of object-oriented design using UML Meiler page-Jones: Pearson Education.
4. Software testing techniques by Boris Beizer, dreamtech.
5. Software Engineering, an Engineering approach- James F. Peters, Witold Pedrycz, John Wiley.
6. Software Engineering principles and practice- Waman S Jawadekar, The Mc Graw-Hill Companies.

19CS2251 - DATABASE MANAGEMENT SYSTEMS LAB

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

- - 3 1.5

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

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.

REFERENCES:

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/

19CS2252 - OPERATING SYSTEMS LAB

B.Tech. II Year II Semester

L T P C
0 0 2 1

Course Objectives

- To write programs in a Linux environment using system calls.
- To implement the scheduling algorithms.
- To implement page replacement algorithms
- To implement file allocation methods.
- To understand and implement IPC mechanisms using named and unnamed pipes.
- To develop solutions for synchronization problems using semaphores.

Course Outcomes

- Ability to develop application programs using system calls in Unix.
- Ability to implement inter process communication between two processes.
- Ability to design and solve synchronization problems.
- Ability to simulate and implement operating system concepts such as scheduling, deadlock management, file management, and memory management.

Use Linux operating system and GNU C compiler.

List of Programs:

1. **a**) Write a shell script that accepts a filename, starting and ending line numbers as arguments and displays all the lines between the given line numbers.
   b) Write a shell script that deletes all lines containing a specific word in one or more files supplied as arguments to it.
   c) Write a shell script that displays a list of all the files in the current directory to which the user has read, write and execute permissions.
2. **a**) Write a shell script that receives any number of filenames as arguments and checks if every argument supplied is a file or a directory, and reports accordingly. Whenever the argument is a file, the number of lines on it is also reported.
   b) Write a shell script that accepts a list of filenames as its argument, counts and reports the accuracy of each word that is present in the first argument file on another argument file.
   c) Write a shell script to list all of the directory files in a directory.
3. **a**) Write a shell script to find the factorial of a given integer.
   b) Write an awk script to count the number of lines in a file that do not contain vowels.
   c) Write an awk script to find the number of characters, words and lines in a file.
4. **a**) Write a C program that makes a copy of a file using standard I/O and system calls.
   b) Implement the following LINUX commands in C using system call.
   i) cat ii) mv

c) Write a C program to list files in a directory.

5. **(a)** Write a C program to copy the contents of one file to another file using system calls.

(b) Write C programs to simulate the following CPU scheduling algorithms:
i) FCFS ii) SJF

6. **a**)Write C programs to simulate the following CPU scheduling algorithms:
i) Round Robin ii) Priority
b) Write a C program to implement the ls| sort command (Use unnamed Pipe).
7. **a**) Write C programs to implement IPC between two unrelated processes using named pipe.
b) Write a C program to solve the Dining- Philosopher problem using semaphores.
8. Write C programs to simulate the following techniques of memory management:
a) Paging **b**) Segmentation
9. Write C programs to simulate the following page replacement algorithms:
a) FIFO **b**) LRU **c**) LFU
10. Write C programs to simulate the following File allocation methods:
a) Contiguous **b**) Linked **c**) Indexed
11. Write C programs to simulate the following File organization techniques:
a) Single level directory b) Two level **c**) Hierarchical
12. Write a C program to simulate Bankers Algorithm for Deadlock Avoidance.
13. Write a C program to simulate Bankers Algorithm for Deadlock Prevention.

TEXT BOOKS:

1. Operating System Principles - Abraham Silchatz, Peter B. Galvin, Greg Gagne 7th Edition, John Wiley.
2. Advanced Programming in the UNIX Environment by W. Richard Stevens Pearson Education.

REFERENCES:

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.

19DS2253 Java Programming Lab

B.Tech. II Year II 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

LIST OF EXPERIMENTS

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 String Buffer 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 inters 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, sixth edition,H.M.Ditiel
2. Programming with java, M.P.Bhave