Jharkhand University of Technology

Ranchi, 834010

SCHEME OF INSTRUCTION AND SYLLABUS

For B.Tech. Program in

Computer Science Engineering

(Effective from 2024-25)

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

Detailed Draft Syllabus

B.Tech. (Computer Science Engineering)

(III – SEMESTER)

PROBABILITY & STATISTICS

BSC-302

Course Outcomes:

At the end of the course, the student will be able to

CO1	Find mean and variance of a given probability distribution
CO2	Test the hypothesis for small and large samples
CO3	Find the coefficient of correlation and lines of regression
CO4	Understand the characteristics of a queuing model

Syllabus:

Random variables and their distributions:

Introduction to Probability, random variables (discrete and continuous), probability functions, density and distribution functions, mean and variance, special distributions (Binomial, Hyper geometric, Poisson, Uniform, exponential and normal), Chebyshev's inequality, parameter and statistic, estimation of parameters by maximum Likelihood Estimation method.

Testing of Hypothesis:

Testing of Hypothesis, Null and alternative hypothesis, level of significance, one-tailed and two-tailed tests, tests for large samples (tests for single mean, difference of means, single proportion, difference of proportions), tests for small samples (t-test for single mean and difference of means, F-test for comparison of variances), Chi-square test for goodness of fit, analysis of variance (one way classification with the samples of equal and unequal sizes), Karl Pearson coefficient of correlation, lines of regression.

Queuing theory:

Concepts, applicability, classification, birth and death process, Poisson queues, Characteristics of queuing models - single server (with finite and infinite capacities) model, multiple server (with infinite capacity only) model.

Text Books / Reference Books / Online Resources:

1. R. A. Johnson, Miller and Freund's "Probability and Statistics for Engineers", Pearson Publishers, 9th Edition, 2017
2. John E. Freund, Benjamin M. Perles, "Modern Elementary Statistics", 12th Edition, Pearson, 2013
3. Hamdy A. Taha, "Operations Research: An Introduction", Pearson, 2017, Tenth Edition
4. S.C.Gupta and V.K.Kapoor, "Fundamentals of Mathematical Statistics", 12th Edition, S.Chand & Co, 2020
5. Kantiswarup, P.K.Gupta and Manmohan Singh, "Operations Research", Sultan Chand & Sons, 2014

DATA STRUCTURES AND ALGORITHMS

CSE-301

COURSE OBJECTIVES

● This course aims to provide the students an in-depth understanding of structure and implementation of the common data structures used in computer science.
● It imparts the ability to solve problems by choosing and applying the right data structures.
● It also imparts the ability to improve the efficiency of programs by applying the right data structures.

COURSE OUTCOMES:-

CO1: Understand the concept and functionalities of Data Structures and be able to implement them efficiently
CO2: Identify and apply appropriate data structures and their libraries to solve problems and improve their efficiency
CO3: Analyze the complexity of data structures and associated algorithms
CO4: Analyze the impact of various implementation and design choices on the data structure performance
CO5: Conceptualize and build data structures based on application needs.

SYLLABUS:-

UNIT 1 REFRESHER OF DATA STRUCTURES: - Abstract Data Types and Data Structures - Principles, and Patterns. Basic complexity analysis – Best, Worst, and Average Cases - Asymptotic Analysis -Analyzing Programs – Space Bounds, recursionlinear, binary, and multiple recursions. -Sorting and Selection – Linear Sorting –Divide and Conquer based sorting – Analysis using Recurrence Tree based Method - Merge Sort - Quick Sort - Studying Sorting through an Algorithmic Lens. Arrays, Linked Lists and Recursion: Using Arrays - Lists - Array based List Implementation – Linked Lists – LL ADT – Singly Linked List – Doubly Linked List – Circular Linked List Stacks and Queues: Stack ADT - Array based Stacks, Linked Stacks – Implementing Recursion using Stacks, Stack Applications. Queues - ADT, Array based Queue, Linked Queue, Double-ended queue, Circular queue, applications.

UNIT 2 TREES: Tree Definition and Properties – Tree ADT - Basic tree traversals - Binary tree - Data structure for representing trees – Linked Structure for Binary Tree – Array based implementation. Priority queues: ADT – Implementing Priority Queue using List – Heaps. Maps and Dictionaries: Map ADT – List based Implementation – Hash Tables - Dictionary ADT. Skip Lists - Implementation - Complexity.

UNIT 3 SEARCH TREES:- – Binary search tree, AVL tree, Trees – Segment Trees - B-Trees. Implementation. External Memory Sorting and Searching. Graphs: ADT- Data structure for graphs - Graph traversal- Transitive Closure- Directed Acyclic graphs - Weighted graphs – Shortest Paths - Minimum spanning tree – Greedy Methods for MST.

TEXT BOOK(S) :-

1.Goodrich MT, Tamassia R, Goldwasser MH. Data structures and algorithms in Python. John Wiley & Sons Ltd; 2013.

REFERENCE(S):-

1.Goodrich MT, Tamassia R, Goldwasser MH. Data structures and algorithms in Java. Fifth edition, John Wiley & Sons Ltd; 2010.
2. Tremblay JP, Sorenson PG. An introduction to data structures with applications. Second Edition, Tata McGrawHill; 2002.
3.Shaffer CA. Data Structures and Algorithm Analysis. Third Edition, Dover Publications; 2012.

DIGITAL ELECTRONICS AND SYSTEMS

CSE-302

COURSE OBJECTIVES :-

● To understand the fundamentals of Boolean Logic and the building blocks of digital circuits
● To introduce the abstraction of simple practical problems into Boolean Logic and their efficient implementation and to introduce the fundamentals of design with combinational and sequential subsystems.

COURSE OUTCOMES:-

CO1: Able to frame Boolean equations for solving a simple real-life Coure and realize them using gate-level building blocks
CO2: Able to apply minimization techniques for efficient Boolean logic implementation
CO3: Able to realize digital blocks using combinational and sequential subsystems
CO4: Able to design using state machine descriptions for practical real-life engineering problems.

SYLLABUS

UNIT 1 INTRODUCTION TO LOGIC CIRCUITS: Variables and functions, inversion - Truth tables - Logic gates and Networks - Boolean algebra - Synthesis using gates - Design examples - Optimized implementation of logic functions: Karnaugh map - Strategy for minimization - Minimization of product of sums forms - Incompletely specified functions - Multiple output circuits - Tabular method for minimization - Number representation and arithmetic circuits: Addition of unsigned numbers - Signed numbers - Fast adders.

UNIT 2 COMBINATIONAL CIRCUIT BUILDING BLOCKS: Multiplexers - Decoders - Encoders - Code converters - Arithmetic comparison circuits. Sequential circuit building blocks: Basic latch - Gated SR latch - Gated D latch - Master slave and edge triggered - D flip-flops - T flip-flop - JK flip-flop - Registers - Counters - Reset synchronization - Other types of counters.

UNIT 3 SYNCHRONOUS SEQUENTIAL CIRCUITS: Basic design steps - State assignment problem - Mealy state model - Serial adders - State minimization. Introduction to Asynchronous sequential circuits – Introduction to CMOS logic.

REFERENCE(S) :-

1.Brown S, Vranesic Z. Fundamentals of Digital logic with Verilog Design.Special Indian Edition, Tata McGraw Hill Publishing Company Limited; 2007.
2.Mano MM, Ciletti MD. Digital Design with Introduction to the Verilog HDL.Fifth Edition, Pearson Education; 2015.
3.Wakerly JF. Digital Design Principles and Practices. Fourth Edition, Pearson Education; 2008.
4.Givone DD. Digital Principles and Design. Tata McGraw Hill Publishing Company Limited; 2003.

OBJECT ORIENTED PROGRAMMING

CSE-303

Course Outcomes:

At the end of the course, the student will be able to

CO1	Understand object oriented paradigms: abstraction, encapsulation, inheritance, polymorphism
CO2	Learn java concepts like exception handling, interfaces, object classes and various libraries.
CO3	Design object oriented solutions for real world problems.
CO4	To be able to apply object-oriented concepts in real world applications.

Syllabus:

Object Oriented Thinking – A way of Viewing the World, Computation as Simulation, Messages and Methods; - A Brief History of Object - Oriented Programming - The History of Java, The White Paper Description, OOPs concept.

- Understanding Inheritance – An Intuitive Description of Inheritance, The Base Class Object, Subclass, Subtype, and Substitutability – Forms of Inheritance, Modifiers and Inheritance, The Benefits of Inheritance, Final, this Keyword, Inheritance: Using Super to Call Super Class Constructor, Method Overriding, Dynamic Method Dispatch, Using Abstract Classes, Using Final with Inheritance;

-The Object Class Packages & Interfaces: Packages, Access Protection, Importing Package, Interface, Implementing Interfaces, Variables in Interfaces, Interfaces can be Extended. Exception Handling: Checked, Unchecked Exceptions, Using Try & Catch, Multiple Catch, Throw, Throws, Finally Java's Built in Exceptions, User Defined Exception.

-- Multi-Threading: Java Thread Model, Thread Priorities, Synchronization, creating a Thread, Creating Multiple Threads, Using is Alive () and Join () Wait () & Notify (). String Handling: String Constructors, String Length, Character Extraction, String Comparison, Modifying a String. Java I/O: Classes & Interfaces, Stream Classes, Byte Streams, Character Streams, Serialization, JDBC: Fundamentals, Type I, Type II, Type III, Type IV Drivers. Networking: Basics, Socket Overview, Networking Classes, & Interfaces, TCP/IP Client Sockets, Whois, URL Format, URL Connection, TCP/IP Server Sockets.

- Polymorphism - Varieties of Polymorphism, Polymorphic Variables, Overloading, Overriding, Abstract Methods, Pure Polymorphism; - The AWT – The AWT Class Hierarchy, The Layout Manager, User Interface Components, Panels, Dialogs, The Menu Bar; - Input And Output Streams - Streams versus Readers and Writers, Input Streams, Stream Tokenizer, Output Streams, Object Serialization, Piped Input and Output;

Applets: Basics, Architecture, Skeleton, The HTML APPLET Tag, Passing Parameters to Applets, Applet Context and Show Documents (). Event Handing: Delegation Event Model, Event Classes, Event Listener Interfaces, Adapter Classes. AWT: AWT Classes Window Fundamentals, Component, Container, Panel Window, Frame, Canvas, Creating a Frame Window in an Applet, Working with Graphics, Control Fundamentals, Layout Managers, Handling Events by Extending AWT Components. Core Java API package, reflection, swing applet, icons & labels, text fields, Buttons, combo boxes, tabbed panes, scroll panes, trees, tables exploring Javalanguage: Simple type wrappers, runtime memory management, object fusing clone () and the cloneable interface, thread, thread group, runnable.

TEXT BOOK:

1. Introduction to Java Programming: Liang, Pearson Education, 7th Edition.

2. Java the Complete Reference: Herbert Schildt, TMH, 5th Edition.

REFERENCE BOOKS:

1. Balguruswamy, Programming with Java, TMH.

2. Programming with Java: Bhave&Patekar, Person Eduction.

3. Big Java: Horstman, Willey India, 2nd Edition.

4. Java Programming Advanced Topics: Wigglesworth, Cengage Learning.

DATABASE MANAGEMENT SYSTEMS

CSE-304

Course Objective

- This course aims to understand the concepts of database design, database languages, database-system implementation and maintenance.

Course Outcomes

CO1: Formulate and apply relational algebraic expressions, SQL and PL/SQL statements to query relational databases.
CO2: Design and build ER models for real world databases.
CO3: Design and build a normalized database management system for real world databases.
CO4: Understand and apply the principles of transaction processing and concurrency control.
CO5: To learn different high level databases and selection of right database.

Syllabus

Unit 1

Introduction: Overview of DBMS fundamentals – Overview of Relational Databases and Keys. Relational Data Model: Structure of relational databases – Database schema – Formal Relational Query Languages – Overview of Relational Algebra and Relational Operations. Database Design: Overview of the design process - The E-R Models – Constraints - Removing Redundant Attributes in Entity Sets - E-R Diagrams - Reduction to Relational Schemas - Entity Relationship Design Issues - Extended E-R Features – Alternative E-R Notations – Overview of Unified Modeling Language (UML).

Unit 2

Relational Database Design: Features of Good Relational Designs - Atomic Domains and 1NF - Decomposition using Functional Dependencies: 2NF, 3NF, BCNF and Higher Normal Forms. Functional Dependency Theory - Algorithm for Decomposition – Decomposition using multi-valued dependency: 4NF and 4NF decomposition. Database design process and its issues. SQL: review of SQL – Intermediate SQL – Advanced SQL.

Unit 3

Transactions: Transaction concept – A simple transaction model - Storage structure - Transaction atomicity and durability - Transaction isolation – Serializability – Recoverable schedules, Casecadeless schedules. Concurrency control: Lock-based protocols – Locks, granting of locks, The two-phase locking protocol, implementation of locking, Graph-based protocols. Deadlock handling: Deadlock prevention, Deadlock detection and recovery. Case Study: Different types of high level databases – MongoDB, Hadoop/Hbase, Redis, IBM Cloudant, DynamoDB, Cassandra and CouchDB etc . Tips for choosing the right database for the given problem.

Text Book(s)

Silberschatz A, Korth HF, Sudharshan S. Database System Concepts. Sixth Edition, TMH publishing company limited; 2011.

Reference(s)

Garcia-Molina H, Ullman JD, Widom J. Database System ; The complete book. Second Edition, Pearson Education India, 2011. Elmasri R, Navathe SB. Fundamentals of Database Systems. Fifth Edition, Addison Wesley; 2006. Ramakrishnan R, Gehrke J. Database Management Systems. Third Edition, TMH; 2003.

DATA STRUCTURES LAB

CSE-301P

COURSE OUTCOMES:-

CO1:- Develop ADT for stack and queue applications.
CO2:- Implement tree and graph algorithms.
CO3:- Implement and analyze internal and external sorting algorithms.
CO4:- Design and implement symbol table using hashing technique.

SYLLABUS

1. Write a program to implement stack using arrays and evaluate a given postfix expression
2. Write a program to implement circular queue using arrays
3. Write a program to implement double ended queue (de queue) using arrays
4. Write programs for applications based on stacks and queues.
5. Write programs to implement the following data structures and their applications (a) Single linked list (b) Double linked list
6. Write programs to implement a stack and a queue using linked lists
7. Write a program to create a binary search tree (BST) by considering the keys in given order and perform the following operations on it. (a) Minimum key (b) Maximum key (c) Search for a given key (d) Find predecessor of a node (e) delete a node with given key (f) applications of BST
8. Write a program to construct an AVL tree for the given set of keys. Also write function for deleting a key from the given AVL tree.
9. Write a program to implement hashing with (a) Separate Chaining and (b) Open addressing methods.
10. Implement the following sorting algorithms: (a) Insertion sort (b) Merge sort (c) Quick sort (d) Heap sort (e) Radix sort (f) Shell sort
11. Write programs for implementation of graph traversals by applying: (a) BFS (b) DFS
12. Write programs to find out a minimum spanning tree of graph by applying: (a) Prim's algorithm (b) Kruskal's algorithm c) any other algorithms
13. Write a program to implement Dijkstra's algorithm using priority queue.
14. Write a program to find Euler's path.
15. Write a programs to find Biconnected components and strongly Connected components.
16. Write program for creation, insertion, and printing functions of a Treap
17. Write program for creation, insertion, deletion and printing functions of a Bd-Tree.
18. Write program for creation, insertion, deletion and printing functions of Bd+-Tree, B*-Tree.
19. Write program for creation, insertion, deletion and printing functions of a Trie.
20. Write program for creation, insertion, deletion and printing functions of a C-Trie. Text Books /

REFERENCE BOOKS / ONLINE RESOURCES:-

1. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, "Introduction to Algorithms", Second Edition, PHI, 2009.
2. Mark Allen Weiss, "Data Structures and Algorithm Analysis in C++", Third Edition, Pearson Education, 2006.
3. Ellis Horowitz, Sartaj Sahni and Sanguthevar Rajasekaran, "Fundamentals of Computer Algorithms", Second Edition, Universities Press, 2011.

4. Michael T.Goodrich and Roberto Tamassia, "Algorithm Design: Foundations, Analysis and Internet Examples", Second Edition, Wiley-India, 2006.

OBJECT ORIENTED PROGRAMMING LAB

CSE-303P

Course Outcomes:

At the end of the course, the student will be able to

CO1	Develop programs using objects and inheritance in Java Language.
CO2	Design and implement GUI programs using components in Java Language

Syllabus:

Java Programming:

1. Ball games
2. Cannon game
3. Pinball game
4. Cards game
5. User interface dialogs related programs
6. I/O processing programs.

Text Books / Reference Books / Online Resources:

1. Timothy Budd, "Object Oriented Programming with JAVA", Updated Edition, Pearson Education, 2009.
2. Herbert Schildt, "Java 2 Complete Reference", TMH, 2010.

DIGITAL ELECTRONICS AND SYSTEMS LAB

CSE-302P

COURSE OBJECTIVES :-

- Familiarization of digital components and ICs used as building blocks for realizing larger systems.
- To learn to realize and troubleshoot simple digital circuits using logic gate ICs on the breadboard and verify their truth tables.
- To learn to use off-the-shelf subsystems such as MSI ICs including adders, decoders and multiplexers by appropriately configuring them with the help of datasheet for realizing circuits to solve a practical engineering problem.

COURSE OUTCOMES :-

CO1: Able to identify, configure and use off-the-shelf digital ICs
CO2: Able to realize and troubleshoot combinational and sequential digital circuits
CO3: Able to employ MSI ICs of appropriate configuration for realizing a digital system
CO4: Able to design and implement simple digital system for a real-life problem.

SYLLABUS

Study of Logic Gate ICs, Realization of Boolean functions using logic gate ICs, Truth table based design and implementation of simple real life problems, Implementation of digital systems using MSI building blocks such as adders, multiplexers and decoders, Breadboard realization of synchronous sequential circuits, Digital system design and implementation for a real-life problem.

REFERENCES(S) :-

1.Brown S, Vranesic Z. Fundamentals of Digital logic with Verilog Design.Special Indian Edition, Tata McGraw Hill Publishing Company Limited; 2007.
2.Mano MM, Ciletti MD. Digital Design with Introduction to the Verilog HDL.Fifth Edition, Pearson Education; 2015.
3.Wakerly JF. Digital Design Principles and Practices. Fourth Edition, Pearson Education; 2008.
4.Navas KA. Electronic Lab Manual – Volume 1. Fifth Edition, Prentice Hall of India; 2015

DATABASE MANGEMENT SYSTEMS LAB

CSE-304P

COURSE OBJECTIVES :-

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

CO1	Design and Implement a database schema
CO2	Devise queries using DDL, DML, DCL and TCL commands.
CO3	Develop application programs using PL/SQL
CO4	Design and implement a project using embedded SQL and GUI.
CO5	Apply modified components for performance tuning in open source software.

Syllabus:

Familiarization of Oracle RDBMS, SQL*Plus and Oracle developer.

SQL: query-structure; DDL-create, alter, drop, rename and Truncate; DML-select, insert, update, delete and lock; Set operations- union, intersection and except; join; Aggregate Operations- group-by and having; nested sub-queries and views; DCL-grant and revoke, TCL-Commit, save point, rollback and set transaction.

PL/SQL: Environment, block structure, variables, operators, data types, control structures; Cursors structures- Implicit and Explicit; Bulk statements- Bulk collect into and forall; Exception handling-Compilation and Run-time, user-defined; Stored procedures- creation options, pass-by-value and functions-pass-by-value; Packages-package specification, body, package creation and usage;

Triggers- Data definition language triggers, Data manipulation triggers, Compound, triggers and trigger restrictions; Large objects-CLOB, NCLOB, BLOB and BFILE; Implementation of applications using GUI; group project;

Text Books / Reference Books / Online Resources:

1. James, Paul and Weinberg, Andy Oppel, "*SQL: The Complete Reference*", 3rd Edition, McGraw Hill, 2011.
2. Michael McLaughlin, "*Oracle Database 11g PL/SQL Programming*", Oracle press.

Detailed Draft Syllabus

B.Tech. (Computer Science Engineering)

(IV – SEMESTER)

OPERATING SYSTEMS

CSE-401

Course Objectives

● This course aims at introducing the structure and implementation of modern operating systems, virtual machines and their applications.
● It summarizes techniques for achieving process synchronization and managing resources like memory, CPU, and files and directories in an operation system.
● A study of common algorithms used for both pre-emptive and non-pre-emptive scheduling of tasks in operating systems (such a priority, performance comparison, and fair-share schemes) will be done.
● It gives a broad overview of memory hierarchy and the schemes used by the operating systems to manage storage requirements efficiently.

Course Outcomes

CO1	Distinguish functional architectures of operating systems and file systems
CO2	Develop algorithms for subsystem components
CO3	Design device drivers and multi-threading libraries for an OS
CO4	Develop application programs using UNIX system calls
CO5	Design and solve synchronization problems

Syllabus

Introduction: Batch, iterative, time sharing, multiprocessor, distributed, cluster and real-time systems, UNIX system introduction and commands

Operating system structures: Computer system structure, Network structure, I/O Structure, Storage Structure, Dual mode operation, System components, Operating-System Services, System Calls, System Programs, System structure, Virtual Machines, System Design and Implementation, System Generation

Processes and Threads : Process Concept, Process Scheduling, Operations on Processes, Cooperating Processes, Interprocess Communication, Communication in Client – Server Systems, Multithreading Models, Threading Issues, Pthreads Basic Concepts,

CPU Scheduling: Scheduling Criteria, Scheduling Algorithms, Multiple-Processor Scheduling, Real-Time Scheduling, Algorithm Evaluation, Process Scheduling Models

Process Synchronization: Synchronization Background, the Critical-Section Problem, Synchronization Hardware, Semaphores, Classic Problems of Synchronization, Critical Regions, Monitors, OS Synchronization

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

Memory Management : Memory Management Background, Swapping, Contiguous Memory Allocation, Paging, Segmentation, Segmentation with Paging, Virtual Memory, Demand Paging, Process Creation, Page Replacement, Allocation of Frames, Thrashing, Operating-System Examples, Other Considerations

File System: File Concept, Access Methods, Directory Structure, File-System Mounting, File Sharing, Protection File-System Structure, File-System Implementation, Directory Implementation, Allocation Methods, Free-Space Management, Efficiency and Performance, Recovery, Log-Structured File System, NFS

I/O Systems : Hardware, Application I/O Interface, Kernel I/O Subsystem, Transforming I/O to Hardware Operations, STREAMS, Performance, Disk Structure , Disk Scheduling , Disk Management, Swap-Space Management, RAID Structure , Disk Attachment, Stable-Storage Implementation, Tertiary-Storage Structure

Text Book(s)
Silberschatz A, Gagne G, Galvin PB. Operating system concepts. Tenth Edition, John Wiley and Sons; 2018.

Reference Book(s)

1. *Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, "Operating System Principles", Wiley, 8/e*
2. *Richard Stevens, Stephen Rago, "Advanced Programming in the UNIX Environment", Pearson Education, 2/e*
3. *Deitel HM, Deitel PJ, Choffnes DR. Operating systems. Third Edition, Prentice Hall; 2004.*

4. *Tannenbaum AS. Modern Operating Systems. Fourth Edition, Prentice Hall; 2016.*

MODELLING AND OPTIMIZATION TECHNIQUES

CSE-402

Course Outcomes:

At the end of the course, the student will be able to

CO1	**Prepare and solve linear programming model.**
CO2	**Model transportation and flow through networks and compute optimal parameters**
CO3	**Optimize inventory levels**
CO4	**Solve real life problems using Meta-heuristic techniques**
CO5	**Generate random numbers and random variates**
CO6	**Verify and validate simulation models**

Syllabus:

Modelling with linear programming – The Simplex method, Sensitivity Analysis,

Integer linear programming: Branch and Bound technique – Transportation Model and its variants,

Network Model: CPM and PERT - Deterministic and non-deterministic inventory models.

Heuristic and Meta Heuristic Programming : Simulated Annealing, Genetic Algorithm, Particle warm Optimization algorithm and Teaching learning based optimization algorithm - Non Linear Programming algorithms**.** Introduction to Quadratic Programming, Constrained Optimization Problem Solving, Convex Optimization Methods.

Simulation Modelling – Random number generation, Random variate generation – Verification and Validation of simulation models, Simulation of Computer Systems and Computer Networks.

Text Books / Reference Books / Online Resources:

1. Hamdy A Taha – “Operations Research-An Introduction”, 9th Ed, Pearson, 2017 (Chs 1-8, 12, 14, 17)
2. Jerry Banks, Hon S Carson, Barry L Nelson, David M Nicol, “Discrete Event Simulation", 5th Ed, Pearson, 2010 (Chs 8 – 12, 14, 15)
3. NPTEL Video Lecturers for Meta Heuristic Techniques by Prof. Prakash Kotecha (IIT Guwahati)

DESIGN AND ANALYSIS OF ALGORITHMS

CSE-403

Course Objectives

- This course aims to provide the fundamentals of algorithm design and analysis, specifically in terms of algorithm design techniques, application of these design techniques for real-world problem solving and analysis of complexity and correctness of algorithms.

Course Outcomes

CO1:	Evaluate the correctness and analyze complexity of algorithms.
CO2	Understand and implement various algorithmic design techniques and solve classical problems
CO3	Design solutions for real world problems by identifying, applying and implementing appropriate design techniques.
CO4	Design solutions for real world problem by mapping to classical problems
CO5	Analyze the impact of various implementation choices on the algorithm complexity

-

Syllabus

Unit 1

Introduction and Review- Algorithms vs. programs. Flow charts and pseudo code, Rate of growth of functions. Review of Asymptotic notation: motivation and types of notations. Recurrence relations and methods to solve them: Recursion, tree, substitution, Master Method. Review of Sorting: Bubble –Insertion – Selection – Bucket – Heap, Comparison of sorting algorithms, Applications. Graph Algorithms – Graph Traversal: Applications of BFS: distance, connectivity and connected components and cycles in undirected graphs. Applications of DFS: Topological sort, cycles in directed graphs, Biconnected Components and Strong Connectivity. Path algorithms: Shortest path algorithms (along with analysis) SSSP: Bellman Ford. APSP: Floyd Warshall's. Review of Minimum Spanning Tree (with analysis and applications).

Unit 2

Divide and Conquer: Merge sort and Binary search type strategies, Pivot based strategies – Long integer multiplication – Maximum sub array sum - Closest Pair problem etc as examples. Greedy Algorithm - Introduction to the method, Fractional Knapsack problem, Task Scheduling Problem, Huffman coding etc as examples. Dynamic Programming: Introduction to the method, Fibonacci numbers, 0-1 Knapsack problem, Matrix chain multiplication problem, Longest Common Subsequence, and other problems including problems incorporating combinatorics as examples.

Unit 3

Backtracking, Branch and Bound 0-1 Knapsack, N- Queen problem, subset sum as some examples. String Matching: Rabin Karp, Boyer Moore, KMP. Network Flow and Matching: Flow Algorithms Maximum Flow – Cuts Maximum Bipartite Matching. Introduction to NP class: Definitions P, NP, NP complete, NP hard, Examples of P and NP.

Text Book(s)

Thomas H Cormen, Charles E Leiserson, Ronald L Rivest and Clifford Stein. Introduction to Algorithms. Third Edition, Prentice Hall of India Private Limited; 2009.

Reference(s)

1. Michael T Goodrich and Roberto Tamassia. Algorithm Design Foundations - Analysis and Internet Examples. John Wiley and Sons, 2007.
2. Dasgupta S, Papadimitriou C and Vazirani U. Algorithms. Tata McGraw-Hill; 2009.
3. Jon Kleinberg, Eva Tardos. Algorithm Design. First Edition, Pearson Education India; 2013

COMPUTER NETWORKS

CSE-404

Course Outcomes:

At the end of the course, the student will be able to

CO1	Understand OSI and TCP/IP models
CO2	Analyze MAC layer protocols and LAN technologies
CO3	Design applications using internet protocols
CO4	Implement routing and congestion control algorithms
CO5	Develop application layer protocols

Syllabus:

Introduction – network architecture - protocol implementation issues - network design. Reference models- The OSI Reference Model- the TCP/IP Model - A Comparison of the OSI and TCP/IP Models

Datalink Layer-Ethernet, Token ring, wireless LANs-Issues with data link Protocols-Encoding framing and error detection and correction-sliding window Protocol-Medium access control

Network layer – network layer design issues - Routing algorithms - Congestion control algorithms –Internetworking - The network layer in the internet - Internet Protocol (IP) - Unicast, multicast, and inter domain routing

Transport layer - Elements of transport protocol - Congestion control – The Internet’s Transmission Control Protocol (TCP) - Remote Procedure Call (RPC) – Implementation semantics of RPC – BSD, sockets - client-server applications

Application layer - Domain name server – Simple Mail Transfer Protocol – File Transfer Protocol -World wide web - Hypertext transfer protocol -Presentation formatting and data compression Introduction to Network security - Web Services architectures for developing new application protocols.

Text Books / Reference Books / Online Resources:

1. Larry L Peterson, Bruce S Davis, "Computer Networks", 5th Edition, Elsevier, 2012.

2. Andrew S. Tanenbaum, David J Wetherall, "Computer Networks", 5th Edition, Pearson Edu, 2010.

ADVANCED PROGRAMMING PRACTICE

CSE-405

Course Objectives

● Primary objective of this course is to introduce advanced programming concepts such as Object oriented paradigm, advanced pointers etc

● This course focuses on learning Python and C++ with an emphasize on ADT and STL usage for implementing data structures.

Course Outcomes

CO1: Understand the static object oriented programming concepts and thereby to understand a given program.
CO2: Understand the dynamic object oriented programming concepts and thereby to understand a given program.
CO3: Implement ADT in static and dynamic object oriented paradigm.
CO4: Analyze the similarities, differences and code efficiency among object oriented programming languages.
CO5: Develop computer programs that implement suitable algorithms for given problem scenario and applications.

Syllabus

Unit 1

Overview of Object Oriented Paradigm, Programming in C++: Objects as a group of variables, Classes as a named group of methods and data, Morphing from structures to classes, Input and Output, Access Specifiers, Member functions: Accessor, Mutator and Auxiliary, Constructors and Destructors, New and Delete Operators, Overloading, Inheritance: Handling Access and Specialization through Overriding, Polymorphism: Virtual Functions, Abstract Class and Virtual Function Tables.

Unit 2

Revisiting Pointers: Pointers to Pointers, Pointers and String Array, Void Pointers and Function Pointers, Standard Template Library, Implementation of Stack, Queue, Hash Table and Linked Lists with STL.

Basic Python: Multi-paradigm language, Data Types and Variables, Indentation, Input and Output statements, Lists and Strings, Deep and Shallow Copy, Tuples and Dictionaries, Set and Frozen Sets, Control Statements and Loops, Iterators and Iterable, Functions, Recursion and Parameter Passing, Namespaces and Variable Scope, Exception Handling.

Unit 3

Object Oriented Concepts in Python: Class, Instance Attributes, Getters, Setters, Inheritance, Multiple Inheritance, Magic Methods and Operator Overloading, Class Creation, Slots, Meta Classes and Abstract Classes, Implementation of Stack, Queue, Hash Table and Linked Lists.

Text Book(s)

1. Stroustrup B. Programming: principles and practice using C++. Second edition, Addison Wesley; 2014.
2. Charles R. Severance. Python for Everybody: Exploring Data Using Python 3, Charles Severance; 2016.

Reference(s)

1. Guttag J. Introduction to Computation and Programming Using Python: With Application to Understanding Data. Second Edition. MIT Press; 2016.
2. Gaddis T. Starting Out with Python. Third Edition, Pearson; 2014.
3. Lambert KA. Fundamentals of Python: first programs. Second Edition, Cengage Learning; 2018.
4. Downey AB. Think Python: How to Think Like a Computer Scientist. O'Reilly Media; 2012.

DESIGN AND ANALYSIS OF ALGORITHM LAB

CSE-403P

COURSE OBJECTIVES: -

This course will enable students to :-

- Design and implement various algorithms in JAVA
- Employ various design strategies for problem solving.
- Measure and compare the performance of different algorithms.

COURSE OUTCOMES:-

The students should be able to:-

- Design algorithms using appropriate design techniques (brute-force, greedy, dynamic programming, etc.)
- Implement a variety of algorithms such assorting, graph related, combinatorial, etc., in a high level language.
- Analyze and compare the performance of algorithms using language features.
- Apply and implement learned algorithm design techniques and data structures to solvereal-world problems.

DESCRIPTION

Design, develop, and implement the specified algorithms for the following problems using Java language under LINUX /Windows environment. Net beans/Eclipse IDE tool can be used for development and demonstration.

LIST OF EXPERIEMENTS

1a) Create a Java Class Called Student with the following details as variables within it

(i) USN

(ii) Name

(iii) Branch

(iv) Phone

Write a Java program to create n Student objects and print the USN, Name, Branch, and Phone of these objects with suitable headings.

1b) Write a Java program to implement the Stack using arrays. Write Push(), Pop(), and Display() methods to demonstrate its working.

2a) Design a super class called Staff with details as Staff Id, Name, Phone, Salary. Extend this class by writing three subclasses namely Teaching (domain, publications), Technical (skills), and Contract (period). Write a Java program to read and display at least 3 staff objects of all three categories.

2b) Write a Java class called Customer to store their name and date_of_birth. The date_of_birth format should be dd/mm/yyyy. Write methods to read customer data as and display as using StringTokenizer class considering the delimiter character as "/".

3a) Write a Java program to read two integers a and b. Compute a/b and print, when b is not zero. Raise an exception when b is equal to zero.

3b) Write a Java program that implements a multi-thread application that has three threads. First thread generates a random integer for every 1 second; second thread computes the square of the number and prints; third thread will print the value of cube of the number.

4) Sort a given set of n integer elements using Quick Sort method and compute its time complexity. Run the program for varied values of n> 5000 and record the time taken to sort. Plot a graph of the time taken versus non graph sheet. The elements can be read from a file or can be generated using the random number generator. Demonstrate using Java how the divideand-conquer method works along with its time complexity analysis: worst case, average case and best case.

5)Sort a given set of n integer elements using Merge Sort method and compute its time complexity. Run the program for varied values of n> 5000, and record the time taken to sort. Plot a graph of the time taken versus non graph sheet. The elements can be read from a file or can be generated using the random number generator. Demonstrate using Java how the divide- and-conquer method works along with its time complexity analysis: worst case, average case and best case.

6) Implement in Java, the 0/1 Knapsack problem using (a) Dynamic Programming method (b) Greedy method.

7) From a given vertex in a weighted connected graph, find shortest paths to other vertices using Dijkstra's algorithm. Write the program in Java.

8)Find Minimum Cost Spanning Tree of a given connected undirected graph using Kruskal'salgorithm. Use Union-Find algorithms in your program.

9) Find Minimum Cost Spanning Tree of a given connected undirected graph using Prim's algorithm.

10)Write Java programs to

(a) Implement All-Pairs Shortest Paths problem using Floyd's algorithm.

(b) Implement Travelling Sales Person problem using Dynamic programming.

11) Design and implement in Java to find a subset of a given set S = {Sl, S2,.....,Sn} of n positive integers whose SUM is equal to a given positive integer d. For example, if S ={1,2, 5, 6, 8} and d= 9, there are two solutions {1,2,6}and {1,8}. Display a suitable message,if the given problem instance doesn't have a solution.

12) Design and implement in Java to find all Hamiltonian Cycles in a connected undirected Graph G of n vertices using backtracking principle.

Operating Systems Lab

CSE-401P

Course Outcomes:

At the end of the course, the student will be able to

CO1	Implement elementary UNIX system commands
CO2	Develop programs to test synchronization problems
CO3	Design and develop user level thread library
CO4	Design and implement file system.

Syllabus:

1. Write Command Interpreter Programs which accepts some basic Unix commands and displays the appropriate result. Each student should write programs for at least six commands.
2. Study the concept of Signals and write a program for Context Switching between two processes using alarm signals.
3. Study pthreads and implement the following: Write a program which shows the performance improvement in using threads as compared with process.(Examples like Matrix Multiplication, Hyper quicksort, Merge sort, Traveling Sales Person problem)
4. Create your own thread library, which has the features of pthread library by using appropriate system calls (UContext related calls). Containing functionality for creation, termination of threads with simple round robin scheduling algorithm and synchronization features.
5. Implement all CPU Scheduling Algorithms using your thread library
6. Study the concept of Synchronization and implement the classical synchronization problems using Semaphores, Message queues and shared memory (minimum of 3 problems)
7. A complete file system implementation inside a disk image file.

Text Books / Reference Books / Online Resources:

1. Richard Stevens, Stephen Rago, "Advanced Programming in the UNIX Environment", Pearson Education, 2/e

Advanced Programming Laboratory

CSE-405P

Objectives

- To explore the features of object oriented programming.
- To focus programming rather on programming language.
- To understand the OS internals.

Outcomes

- Ability to develop shell scripts for various applications.
- Gaining knowledge about OS internals.
- Understanding Object oriented concepts and developing software modules.

SYLLABUS

- Exercises using Linux tools – Grep, awk, tr
- Exercises using system calls
- Exercises in Python
- Exercises in C++/ Java
- Introduction to Java
- Java Applets; the Java Development Kit (JDK) (in Lab);
- Exception Handling
- Graphical User Interface (GUI),
- Multithreading,
- Java Network Programming:
- URL class, URL Connection class;
- InetAddress class, Socket class
- Client Server Programming;
- Remote method invocation (Java.rmi package),
- Database manipulation in Java

Reference Books

- Arnold Robbins, Nelson H. F. Beebe, Classic Shell Scripting, O’Reilly Media 2005
- H. Schildt Java: The Complete Reference, Eighth Edition, McGraw-Hill Education (India) Pvt. Limited, 2011.
- H. Schildt C++: The Complete Reference, Fourth Edition, McGraw-Hill Education (India) Pvt Limited, 2003.
- Mark Lutz Learning Python, 3rd Edition, O’Reilly Media, 2007