BRAINWARE UNIVERSITY
SCHOOL OF ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
Bachelor of Technology in Computer Science & Engineering - Artificial Intelligence & Machine Learning 2023

SEMESTER – IV

Course Code	Course Name	Course Type	Hours per week			Credits	Total Marks
			L	T	P		
HSMCM401	Design Thinking for Entrepreneurship and Startups	HS	3	0	0	3	100
PCC-CSM401	Operating Systems	PC	3	0	0	3	100
PCC-CSM402	Design and Analysis of Algorithms	PC	3	0	0	3	100
PCC-CSM403	Object Oriented Programming	PC	3	0	0	3	100
PCC-CSM404	Formal Language & Automata Theory	PC	3	0	0	3	100
PCC-CSM405	Discrete Mathematics	PC	4	0	0	4	100
PCC-CSM491	Operating Systems Lab	PC	0	0	3	1.5	100
PCC-CSM492	Design and Analysis of Algorithms Lab	PC	0	0	3	1.5	100
PCC-CSM493	Object Oriented Programming Lab	PC	0	0	3	1.5	100
Total						**23.5**	**900**

Course Code: HSMCM401

Course Name: Design Thinking for Entrepreneurship and Startups

Contact: 3L

Credit: 3

Allotted Hour: 45L

Course Objective:

BRAINWARE UNIVERSITY
SCHOOL OF ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
Bachelor of Technology in Computer Science & Engineering - Artificial Intelligence & Machine Learning 2023

The objective of this Course is to provide the new ways of creative thinking and Learn the innovation cycle of Design Thinking process for developing innovative products which are useful for a student in preparing for an engineering career. Acquiring Entrepreneurial spirit and resourcefulness. Understanding the concept and process of entrepreneurship - its contribution and role in the growth and development of individuals and the nation. Acquiring entrepreneurial quality, competency, and motivation.

Pre-requisite(s)*:* Good knowledge of Fundamentals of Software Engineering and Management.

Course Outcome: After the completion of the course, students would be able to:

CO1: Describe entrepreneurship and visualize various designing processes for global business structures.

CO2: Construct innovative thinking, and design the creative product and services per se.

CO3: Evaluate market needs, and focus on prototype development for customers.

CO4: Develop ideas and prepare business plans.

CO5: Recognize and analyze implications of IPR at global context.

Module I: **[5H]**

Introduction to Entrepreneurship, Design Thinking: Definition of Design Thinking and Entrepreneurship, Need for Design Thinking and Entrepreneurship, Objective of Design Thinking, Concepts & Brainstorming, Stages of Design Thinking Process (explain with examples) – Empathize, Define, Ideate, Prototype, Test. Traits of an entrepreneur, Intrapreneurship, Motivation. Types of Business Structures.

Module II: **[4H]**

Being Ingenious & Fixing Problem: Understanding Creative thinking process, Understanding Problem Solving, Testing Creative Problem Solving.

Module III: **[4H]**

Process of Product Design: Process of Engineering Product Design, Design Thinking Approach, Stages of Product Design, Examples of best product designs and functions, Assignment – Engineering Product Design.

Module IV: **[6H]**

Prototyping & Testing: What is Prototype? Why Prototype? Rapid Prototype Development process, Testing, Sample Example, Test Group Marketing.

Module V: **[4H]**

Design Thinking & Customer Centricity: Practical Examples of Customer Challenges, Use of Design Thinking to Enhance Customer Experience, Parameters of Product experience, Alignment of Customer Expectations with Product Design.

Module VI: **[6H]**

Business Ideas and their implementation: Discovering ideas and visualizing the business, Activity map, Business Plan.

Module VII: **[10H]**

Managing the Business: Idea to Start-up: Market Analysis – Identifying the target market, Competition evaluation and Strategy Development. Financing and Protection of Ideas: Financing methods available for start-ups in India Management: Recruitment and management of talent.

Module VIII: **[6H]**

Intellectual Property Rights (IPR): Introduction and the economics behind development of IPR: Business Perspective, IPR in India – Genesis and Development. International Context Concept of IP Management, Use in marketing.

Text books:

1. “Engineering Design”, Cengage learning (International edition), John.R.Karsnitz, Stephen O’Brien and John P. Hutchinson, Second Edition, 2013.
2. "The Design of Business: Why Design Thinking is the Next Competitive Advantage", Roger Martin, Harvard Business Press, 2009.
3. "Design Thinking: Understand – Improve – Apply", Hasso Plattner, Christoph Meinel and Larry Leifer (eds), Springer, 2011.
4. "Design Thinking for Strategic Innovation: What They Can't Teach You at Business or Design School", Idris Mootee, John Wiley & Sons 2013.

5. The Startup Owner's Manual: The Step-by-Step Guide for Building a Great Company, Steve Blank and Bob Dorf,K & S Ranch ISBN – 978-0984999392.
6. The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses, Eric Ries, Penguin UK ISBN – 978-0670921607.
7. Demand: Creating What People Love Before They Know They Want It, Adrian J. Slywotzky with Karl Weber, Headline Book Publishing ISBN – 978-0755388974.

BRAINWARE UNIVERSITY
SCHOOL OF ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
Bachelor of Technology in Computer Science & Engineering - Artificial Intelligence & Machine Learning 2023

Course Code: PCC-CSM401

Course Name: Operating Systems

Contact: 3L

Credit: 3

Allotted Hour: 45L

Course Objective:

This course has two components: a theory component to teach the concepts and principles that underlie modern operating systems, and a practice component to relate theoretical principles with operating system implementation. In the theory component, the student should understand the basic concept of operating system, various types of operating systems and their structures. Different types of services and functions that are provided by the operating systems, Basic concept of process and thread, Concept about the processes management. Students should also understand the services provided by an operating system, the structure and organization of the file system. The processes are synchronized and scheduled, different approaches to memory management. Students should be able to use system calls for managing processes, memory and the file system and understand the data structures and algorithms used to implement an OS.

Pre-requisite(s): Basic programming logic, Concept of Data Structure & Algorithm, Basic mathematics and digital electronics.

Course Outcome: After the completion of the course, students would be able to:

CO1: Define the role of Operating System and explain the design of control unit.

CO2: Investigate and analyze CPU Scheduling, Synchronization and Deadlock Handling to evaluate the performance of the system.

CO3: Describe the role of paging, segmentation and virtual memory in operating systems to solve logical problems.

CO4: Explain protection and security and also the Comparison of UNIX and Windows based OS.

CO5: Understand I/O system, Device Management Policies & Secondary Storage Structure and evaluate various Disk Scheduling Algorithms.

Module I: **[8H]**

Introduction: Basic Concept of Operating Systems, Definition, Roles of Operating System, User & System View, Computer System Architecture: Single processor and Multi-processor System, Graceful degradation, Types of Operating Systems, Multiprogramming vs. Multitasking, Dual mode operations in OS, Operating System Services, Resource Allocation, System Calls, Operating System Structure: Simple, Complex & Layered Structure, Types of kernel in Operating Systems, Concept of Virtual Machine. Basic concept of Tizen Operating system. Implementation of Tizen OS in IoT.

Module II: **[8H]**

Processes: Definition, Process Relationship, Different states of a Process, Process State transitions, Process Control Block (PCB), Context switching.

Process Scheduling: Foundation and Scheduling objectives, Types of Schedulers, Scheduling criteria: CPU utilization, Throughput, Turnaround Time, Waiting Time, and Response Time.

Scheduling algorithms: Pre-emptive and Non pre-emptive, FCFS, SJF, Priority & Round Robin CPU Scheduling Algorithm, Multilevel Queue and Multilevel Feedback Queue Scheduling.

Thread: Definition, Various states, Benefits of threads, Types of threads, Concept of multi threads.

Real Time scheduling: RM and EDF.

Module III: **[8H]**

Inter-process Communication: Independent vs. cooperative process, advantages and disadvantages of cooperative process**.** Share memory vs. message passing concept**.**

Process Synchronization: Definition and utilization. Concept of Critical Section, Race Conditions, Critical section problem and its solution. Mutual Exclusion, Bounded Buffer, Hardware Solution, Peterson's Solution, The Producer Consumer Problem, Semaphores, wait() & signal(), Event Counters, Monitors, Record-based vs. Binary Semaphores, Classical IPC Problems: Reader's & Writer Problem, Dining Philosopher Problem etc.

Module IV: **[6H]**

Deadlocks: Definition, Necessary and sufficient conditions for Deadlock: Mutual Exclusion, Hold and wait, No preemption, Circular wait; Resource allocation graph and its component; Deadlock Handling: Deadlock Prevention, Deadlock Avoidance: Banker's algorithm, Deadlock detection and Recovery**.** Resource Request Algorithm**.** Wait for Graph.

Module V: [10H]

Memory Management: Basic concept, various types of address binding, dynamic loading and linking, overlays, Logical vs. Physical address map, swapping, Single & Multiple Partition Allocation Schemes with example, Memory Allocation Strategy Example, Compaction and Coalescing with Example, Memory allocation: Contiguous Memory allocation–Fixed and variable partition– Internal and External fragmentation and Compaction; Thrashing in Operating System, Memory Allocation Algorithm.

Paging: Principle of operation – Page allocation – Hardware support for paging, Protection and sharing, Disadvantages of paging. Multilevel Paging and Inverted Page Table, Memory Management Using Paging, Memory Management Using Segmentation.

Virtual Memory: Basics of Virtual Memory – Hardware and control structures – Locality of reference, Page fault , Working Set, Dirty page/Dirty bit – Demand paging, Page Replacement algorithms: Optimal, First in First Out (FIFO), Least Recently used (LRU) etc.

Module VI: [5H]

I/O Hardware: I/O devices, Device controllers, Direct memory access Principles of I/O Software: Goals of Interrupt handlers, Device drivers, Device independent I/O software, Secondary-Storage Structure: Disk structure. Input Output Management, Input Output Kernel Subsystem, Interrupts.

File Management: Concept of File, Access methods, File types, File operation, Directory structure, File System structure, Allocation methods (contiguous, linked, indexed), Free-space management (bit vector, linked list, grouping), directory implementation (linear list, hash table), efficiency and performance. File System I-Node.

Disk Management: Disk structure, Disk scheduling - FCFS, SSTF, SCAN, C-SCAN, C-LOOK, Disk formatting, Boot-block, Bad blocks.

Text books:

1. Operating System Concepts Essentials, 9th Edition by Avi Silberschatz, Peter Galvin, Greg Gagne, Wiley Asia Student Edition.
2. Operating Systems: Internals and Design Principles, William Stallings, Prentice Hall of India. 5th Edition.

Reference books:

1. Operating System: A Design-oriented Approach, Charles Crowley, Irwin Publishing, 1st Edition.
2. Operating Systems: A Modern Perspective, Gary J. Nutt, Addison-Wesley, 2nd Edition.
3. Design of the Unix Operating Systems, by Maurice Bach, Prentice-Hall of India, 8th Edition.
4. Understanding the Linux Kernel, Daniel P. Bovet, Marco Cesati, O'Reilly and Associates, 3rd Edition.

BRAINWARE UNIVERSITY
SCHOOL OF ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
Bachelor of Technology in Computer Science & Engineering - Artificial Intelligence & Machine Learning 2023

Course Code: PCC-CSM402

Course Name: Design and Analysis of Algorithms

Contact: 3L

Credit: 3

Allotted Hour: 45L

Course Objective:

The objective of this course is to provide a mathematical foundation for analyzing and proving the efficiency of an algorithm. This also makes the students familiar with main thrusts of work in algorithms to give some context for formulating and seeking known solutions to an algorithmic problem.

Pre-requisite(s): Programming Knowledge, Fundamental Mathematics.

Course Outcome: After the completion of the course, students would be able to:

CO1: Define and analyze the complexity of different algorithms.

CO2: Explain different types of algorithms and compare their behavior.

CO3: Discuss and analyze the concept of graph traversal techniques.

CO4: Discuss and analyze the algorithm and differentiate among P, NP, NP hard and NP complete class.

CO5: Define and evaluate Approximation and Randomized algorithm.

CO6: Discuss different algorithm based on case studies to solve real life problems.

Module I: **[6H]**

Introduction: Stages of algorithm development for solving a problem: Describing the problem, identifying a suitable technique, Design of an algorithm, Proof of Correctness of the algorithm, Asymptotic Notation, Recurrence Relation, Methods to solve Recurrence relation(Master Theorem, Substitution method, Recurrence tree).

Module II: **[12H]**

Algorithm Design Techniques: Brute force techniques – Traveling Salesman Problem, Divide and Conquer - Finding a maximum and minimum in a given array -Matrix multiplication: Strassen's algorithm, Greedy techniques Huffman Codes and Data Compression -Fractional Knapsack problem, Dynamic programming - O/1 Knapsack problem-Matrix chain multiplication, LCS, Travelling Salesman Problem, Backtracking N-Queens Problem, Knight's Tour on Chess Board.

Module III: **[4H]**

String Matching Algorithms: Naïve String Matching Algorithms, KMP algorithm, Rabin-Karp Algorithm.

Module IV: **[8H]**

Graph Algorithms: All pair shortest path – Floyd-Warshall Algorithm. Network Flows - Flow Networks, Maximum Flows – Ford-Fulkerson Algorithm, Push Relabel Algorithm, Minimum Cost Flows – Cycle Cancelling Algorithm.

Module V: **[6H]**

Complexity Classes: The Class P, The Class NP, Reducibility and NP-completeness – SAT (without proof), 3-SAT, Vertex Cover, Independent Set, Maximum Clique.

Module VI: **[6H]**

Approximation and Randomized Algorithms: Approximation Algorithms - The set-covering problem – Vertex cover, K-center clustering. Randomized Algorithms - The hiring problem, Finding the global Minimum Cut.

Module VII: **[3H]**

Current Trends.

Text books:

1. Introduction to Algorithms, Thomas H. Cormen, C.E. Leiserson, R L.Rivest and C. Stein, MIT Press Third edition, 2009

Reference books:

1. Algorithm Design, Jon Kleinberg, Éva Tardos, Pearson education, 2014

BRAINWARE UNIVERSITY
SCHOOL OF ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
Bachelor of Technology in Computer Science & Engineering - Artificial Intelligence & Machine Learning 2023

Course Code: PCC-CSM403

Course Name: Object Oriented Programming

Contact: 3L

Credit: 3

Allotted Hour: 45L

Course Objective:

Programming is an essential tool for all the students mainly who are from Computer Science. This course covers the basic concepts of programming like flowchart and algorithm and helps to solve a given problem. The course covers comparison of OOP languages with other conventional languages, features of OOP like encapsulation, abstraction, inheritance etc. It reflects the popularity of Java language due to byte code. The course has a lab component where the students practically solve a problem using Java programming language. The objective of the course is to familiarize the prospective engineers with the basic and important OOP programming language using Java. It aims to equip the students with standard concepts of the language and help them solve a problem using this programming language and also the features that make it more popular.

Pre-requisite(s): Basic programming concept, defining logic for solution of a problem.

Course Outcome: After the completion of the course, students would be able to:

CO1: Recognize the basic ideas of programming and understand the features of OOP.

CO2: Describe and apply some features of OOP like constructor, polymorphism and string class to solve real time problems.

CO3: Explain and Implement different types of reusability properties like inheritance, abstract class, interface, packages and multi-threading.

CO4: Examine and evaluate exception classes, exception handling in Java.

CO5: understand and apply the uses of GUI in Java.

BRAINWARE UNIVERSITY
SCHOOL OF ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
Bachelor of Technology in Computer Science & Engineering - Artificial Intelligence & Machine Learning 2023

Module I: **[7H]**

Object Oriented concepts: Difference between OOP and other conventional programming – advantages and disadvantages. Concept of Class, object, message passing, inheritance, encapsulation, polymorphism.

Module II: **[10H]**

Class & Object properties: Basic concepts of Java programming – advantages of java, byte-code & JVM, datatypes, access specifiers, operators, control statements & loops, array, creation of class, object, constructor, finalize and garbage collection, use of method overloading, this keyword, use of objects as parameter & methods returning objects, call by value & call by reference, static variables & methods, garbage collection, nested &inner classes, basic string handling concepts- String class (charAt(), compareTo(), equals(), equalsIgnoreCase(), indexOf(), length(), substring(), toCharArray(), toLowerCase(), toString(), toUpperCase(), trim(), valueOf() methods) & StringBuffer class (append(), capacity(), charAt(), delete(), deleteCharAt(), ensureCapacity(), getChars(), indexOf(), insert(), length(), setCharAt(), setLength(), substring(), toString() methods), concept of mutable and immutable string, command line arguments, basics of I/O operations –keyboard input using BufferedReader & Scanner classes.

Module III: **[10H]**

Reusability properties: Super class & sub-classes including multilevel hierarchy, process of constructor calling in inheritance, use of super and final keywords with super() method, dynamic method dispatch, use of abstract classes & methods, interfaces. Creation of packages, importing packages, member access for packages.

Module IV: **[9H]**

Exception handling & Multithreading: Basics of Exception handling, different types of exception classes, use of try & catch block with throw, throws & finally, creation of user defined exception classes. Basics of multi threading, main thread, thread life cycle, creation of multiple threads, thread priorities, thread synchronization, inter-thread communication, deadlocks for threads, suspending & resuming threads.

Module V: **[9H]**

GUI: Introduction to AWT programming, Layout and component managers, Event handling, Swing components – JButton, JFrame, etc, Sample swing programs.

Text books:

1. Program Development in Java, Barbara Liskov, Addison-Wesley, 2001.
2. Core Java For Beginners, R.K. Das, Vikas Publishing, 3rd edition, 2013.
3. The complete reference JAVA2, Hervert schildt. TMH.

Reference books:

1. Beginning Java 2, Ivor Horton, Wrox, 5th edition, 2005.
2. Java How to Program, Deitel and Deitel, Pearson, 11th edition, 2017.

BRAINWARE UNIVERSITY
SCHOOL OF ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
Bachelor of Technology in Computer Science & Engineering - Artificial Intelligence & Machine Learning 2023

Course Code: PCC-CSM404

Course Name: Formal Language & Automata Theory

Contact: 3L

Credit: 3

Allotted Hour: 45L

Course Objective:

The objective of this course is to learn the basic concepts of Formal Languages and Automata and classify various types of Machines, Languages and Grammar. This also distinguishes between various normal forms and defines the languages which are regular or context free with the use of pumping lemma.

Pre-requisite(s): Set theory.

Course Outcome: After the completion of the course, students would be able to:

CO1: Explain and manipulate the different concepts in automata theory and formal languages.

CO2: Describe the language accepted by an automata and Prove properties of languages, grammars and automata with rigorously formal mathematical methods.

CO3: Sketch the different type of Automata for different types of Languages and Simplify automata and context-free grammars.

CO4: Distinguish between various normal forms and transform between pushdown automata and contest free grammars to a competent level.

CO5: Define turing machines performing simple tasks and Judge whether certain languages are regular or context free with the use of pumping lemma.

Module I: **[7H]**

Introduction: Alphabet, languages and grammars, productions and derivation, Chomsky hierarchy of languages.

Module II: **[10H]**

Regular languages and finite automata: Regular expressions and languages, deterministic finite automata (DFA) and equivalence with regular expressions, nondeterministic finite automata (NFA) and equivalence with DFA, regular grammars and equivalence with finite automata, properties of regular languages, pumping lemma for regular languages, minimization of finite automata.

Module III: **[10H]**

Context-free languages and pushdown automata: Context-free grammars (CFG) and languages (CFL), Chomsky and Greibach normal forms, nondeterministic pushdown automata (PDA) and equivalence with CFG, parse trees, ambiguity in CFG, pumping lemma for context-free languages, deterministic pushdown automata, closure properties of CFLs.

Module IV: **[8H]**

Context-sensitive languages: Context-sensitive grammars (CSG) and languages, linear bounded automata and equivalence with CSG.

Module V: **[10H]**

Turing machines: The basic model for Turing machines (TM), Turing- recognizable (recursively enumerable) and Turing-decidable (recursive) languages and their closure properties, variants of Turing machines, nondeterministic TMs and equivalence with deterministic TMs, unrestricted grammars and equivalence with Turing machines, TMs as enumerators.

Undesirability: Church-Turing thesis, universal Turing machine, the universal and diagonalization languages, reduction between languages and Rice's theorem, undecidable problems about languages.

Text books:

1. Introduction to Automata Theory, Languages, and Computation, John E. Hopcroft, Rajeev Motwani and Jeffrey D. Ullman, Pearson Education Asia.

Reference books:

1. Elements of the Theory of Computation,Harry R. Lewis and Christos H. Papadimitriou,Pearson Education Asia.
2. Introduction to the Theory of Computation, Michael Sipser, PWS Publishing.

3. Introduction to Languages and The Theory of Computation ,John Martin , Tata McGraw Hill.

BRAINWARE UNIVERSITY
SCHOOL OF ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
Bachelor of Technology in Computer Science & Engineering - Artificial Intelligence & Machine Learning 2023

Course Code: PCC-CSM405

Course Name: Discrete Mathematics

Contact: 4L

Credit: 4

Allotted Hour: 60L

Course Objective:

The objective of this course is to familiarize the prospective computer scientists with the techniques of mathematical reasoning, logical thinking, abstract mathematical discrete structures so that they may apply a particular set of mathematical facts in relevant situations.

Pre-requisite(s): Algebra, Geometry and Precalculus.

Course Outcome: After the completion of the course, students would be able to:

CO1: Remember and understand different types of functions, mathematical induction and the greatest common divisor.

CO2: Understand group structures and the knowledge of algebraic structures and solve related computer engineering problems.

CO3: Apply counting techniques and algorithms to solve related computer engineering problems and evaluate the same.

CO4: Understand, apply and analyze different types of graphs to solve various problems of graph theory.

CO5: Understand propositional logic and evaluate mathematical statements logically by using them.

Module I: **[12H]**

Sets, Relation and Function: Operations and Laws of Sets, Cartesian Products, Binary Relation, Partial Ordering Relation, Equivalence Relation, Image of a Set, Sum and Product of Functions, Bijective functions, Inverse and Composite Function, Size of a Set, Finite and infinite Sets, Countable and Uncountable Sets, Cantor's diagonal argument and The Power Set Theorem, Schroeder-Bernstein Theorem, Principles of

Mathematical Induction: The Well-Ordering Principle, Recursive definition, The Division algorithm: Prime Numbers, The Greatest Common Divisor: Euclidean Algorithm, The Fundamental Theorem of Arithmetic.

Module II: **[10H]**

Counting techniques: Basic counting techniques-Inclusion and Exclusion, Pigeon-hole Principle, Permutation and Combination, Generating function and recurrence relations.

Module III: **[10H]**

Propositional Logic: Syntax, Semantics, Validity and Satisfiability, Basic Connectives and Truth Tables, Logical Equivalence: The Laws of Logic, Logical Implication, Rules of Inference, The use of Quantifiers. Mathematical Induction.

Module IV: **[14H]**

Algebraic Structures and Morphism: Algebraic Structures with one Binary Operation, Semi Groups, Monoids, Groups, Congruence Relation and Quotient Structures, Groups, Permutation Groups, Substructures, Normal Subgroups, Algebraic Structures with two Binary Operation, Rings, Integral Domain and Fields. Boolean Algebra and Boolean Ring, Identities of Boolean Algebra, Duality, Representation of Boolean Function, Disjunctive and Conjunctive Normal Form.

Module V: **[14H]**

Graphs and Trees: Graphs and their properties, Degree, Connectivity, Path, Cycle, Sub Graph, Isomorphism, Eulerian and Hamiltonian Walks, Shortest path Algorithm, Trees and spanning trees and related Algorithms.

Text books:

1. “Discrete Mathematics and Its Applications”, Kenneth H. Rosen, McGraw-Hill.
2. “Discrete Mathematics with Applications”, Susanna S Epp, Wadsworth Publishing Co. Inc, 4th edition.
3. “Elements of Discrete Mathematics: a computer oriented approach”, C L Liu and Mohapatra, McGraw Hill, 3rd edition.

Reference books:

1. “Discrete Mathematical Structures and its Application to Computer Science”, J P Trembley, R Manohar, TMG Edition, Tata McGraw-Hill.
2. “Discrete Mathematics”, Norman L Biggs, Oxford University Press, 2nd Edition.
3. “Discrete Mathematics”, Schaum’s Outlines Series, Semyour Lipschutz and Marc Lipson.

BRAINWARE UNIVERSITY
SCHOOL OF ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
Bachelor of Technology in Computer Science & Engineering - Artificial Intelligence & Machine Learning 2023

Course Code: PCC-CSM491

Course Name: Operating Systems Lab

Contact: 3P

Credit: 1.5

Allotted Hour: 45L

Course Objective:

This course has two components: a theory component to teach the concepts and principles that underlie modern operating systems, and a practice component to relate theoretical principles with operating system implementation.

Pre-requisite(s): Basic programming logic, Concept of Data Structure & Algorithm, Basic mathematics and digital electronics.

Course Outcome: After the completion of the course, students would be able to:

CO1: Understand fundamental operating system abstractions like process, threads and apply this knowledge to solve real time problem.

CO2: Analyze different types of Process scheduling algorithms and solve the real time problem.

CO3: Illustrate the semaphore and IPC and solved real time problem using semaphore and some IPC functions.

CO4: Categorize the operating system's resource management techniques, memory management techniques and deadlock management techniques and apply this techniques to solve the real time problem efficiently without any deadlock.

CO5: Evaluate the ability to perform OS tasks in Linux and solve real time problem using shell script.

Module I: **[15H]**

Process: Starting new process, replacing a process image, duplicating a process image, waiting for a process, zombie process.

Scheduling algorithms: Pre-emptive and Non pre-emptive, FCFS, SJF, Priority & Round Robin CPU Scheduling Algorithm, Multilevel Queue and Multilevel Feedback Queue Scheduling.

Module II: **[3H]**

Signal: Signal handling, sending signals, signal interface, signal sets.

Module III: **[9H]**

Semaphore: programming with semaphores (use functions semctl, semget, semop, set_semvalue, del_semvalue, semaphore_p, semaphore_v).

Inter-process communication: pipes (use functions pipe, popen, pclose), named pipes(FIFOs, accessing FIFO).

Module IV: **[6H]**

Deadlocks: Deadlock Prevention, Deadlock Avoidance: Banker's algorithm, Deadlock detection and Recovery**.** Resource Request Algorithm**.** Wait for Graph.

Module V: **[12H]**

Shell programming: creating a script, making a script executable, shell syntax (variables, conditions, control structures, functions, commands).

Text books:

1. Operating System Concepts Essentials, 9th Edition by Avi Silberschatz, Peter Galvin, Greg Gagne, Wiley Asia Student Edition.
2. Operating Systems: Internals and Design Principles, William Stallings, Prentice Hall of India. 5th Edition.

Reference books:

1. Operating System: A Design-oriented Approach, Charles Crowley, Irwin Publishing, 1st Edition.
2. Operating Systems: A Modern Perspective, Gary J. Nutt, Addison-Wesley, 2nd Edition.
3. Design of the Unix Operating Systems, by Maurice Bach, Prentice-Hall of India, 8th Edition.
4. Understanding the Linux Kernel, Daniel P. Bovet, Marco Cesati, O'Reilly and Associates, 3rd Edition.

BRAINWARE UNIVERSITY
SCHOOL OF ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
Bachelor of Technology in Computer Science & Engineering - Artificial Intelligence & Machine Learning 2023

Course Code: PCC-CSM492

Course Name: Design and Analysis of Algorithms Lab

Contact:3P

Credit:1.5

Allotted Hour: 45L

Course Objective:

The objective of this course is to design and implement efficient algorithms for a specified application or problem.

Pre-requisite(s): Programming Knowledge.

Course Outcome: After the completion of the course, students would be able to:

CO1: Solve the problems using different techniques to analyze the complexity of different algorithms.

CO2: Analyze different types of algorithms and compare their behaviour.

CO3: Solve and analyze the concept of graph traversal techniques.

CO4: Compare different types of algorithm after solving their time complexity.

CO5: Evaluate the algorithm and judge the best possible algorithm for a specific application.

CO6: Design an algorithm based on case studies to solve real life problems.

Module I: **[3H]**

QUICK SORT Sort a given set of elements using the quick sort method and determine the time required to sort the elements. Repeat the experiment for different values of n, the number of elements in the 1st to be sorted and plot a graph of the time taken versus n. The elements can be read from a file or can be generated using the random number generator or can be taken as input.

Module II: **[3H]**

Implement merge sort algorithm to sort a given set of elements and determine the time required to sort the elements. Repeat the experiment for different values of n, the number of elements in the list to be sorted and plot a graph of the time taken versus n. The elements can be read from a file or can be generated using the random number generator.

Module III: **[3H]**

WARSHALL'S ALGORITHM: Compute the transitive closure of a given directed graph using Warshall's algorithm.

Module IV: **[3H]**

Implement 0/1 Knapsack problem using Dynamic Programming/Greedy Algorithm.

Module V: **[3H]**

Implement Fractional Knapsack problem using Greedy Algorithm.

Module VI: **[3H]**

SHORTEST PATHS ALGORITHM From a given vertex in a weighted connected graph, find shortest paths to other vertices using Dijkstra's algorithm.

Module VII: **[3H]**

Find Minimum Cost Spanning Tree of a given undirected graph using Kruskal's algorithm.

Module VIII: **[3H]**

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

Module IX: **[3H]**

Print all the nodes reachable from a given starting node in a digraph using BFS method.

Module X: **[3H]**

Print all the nodes reachable from a given starting node in a digraph using DFS method.

Module XI: **[3H]**

ALL PAIRS SHORTEST PATHS Implement All-Pairs Shortest Paths Problem using Floyd's algorithm.

Module XII: **[3H]**

Implement N Queen's problem using Back Tracking.

Module XIII: **[3H]**

Implement Graph Colouring problem using Back Tracking.

Module XIV **[3H]**

Implement any scheme to find the optimal solution for the Traveling Sales Person problem and then solve the same problem instance using any approximation algorithm and determine the error in the approximation.

Module XV: **[3H]**

Find a subset of a given set S = {s1, 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}. A suitable message is to be displayed if the given problem instance doesn't have a solution.

Text books:

1. “Introduction to the Design And Analysis of Algorithms”, Levitin A, Pearson Education.
2. “Introduction to Algorithms”, Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein.

BRAINWARE UNIVERSITY
SCHOOL OF ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
Bachelor of Technology in Computer Science & Engineering - Artificial Intelligence & Machine Learning 2023

Course Code: PCC-CSM493

Course Name: Object Oriented Programming Lab

Contact: 3P

Credit: 1.5

Allotted Hour: 45L

Course Objective:

The objective of this course is to recall the basics of algorithm and logic for solving different basic problems using the features of Java programming language.

Pre-requisite(s): algorithm concept, basic programming logic.

Course Outcome: After the completion of the course, students would be able to:

CO1: Apply the simple knowledge of oops to solve the basic real time problem using java.

CO2: Illustrate the features of polymorphism and String class to solve the real time problem using java.

CO3: Implement the property of inheritance, abstract and interface for solving the real time problem using java.

CO4: Evaluate different types of exception and errors and solve these exception and error by using error handling mechanism in Java.

CO5: Analyze thread and GUI to develop Window application using java.

Module I **[3H]**

Simple Java program with a class and object.

Module II: **[9H]**

Method Overloading, user input using different classes, String methods, StringBuffer methods, command line argument.

Module III: **[15H]**

Constructor**,** Inheritance, abstract class, interface, package.

Module IV: **[9H]**

Exception using try-catch block, user defined exception and thread life cycle.

Module V: **[9H]**

AWT and Swing programming.

Text books:

1. Program Development in Java, Barbara Liskov, Addison-Wesley, 2001.
2. Core Java For Beginners, R.K. Das, Vikas Publishing, 3rd edition, 2013.
3. Beginning Java 2, Ivor Horton, Wrox, 5th edition, 2005.
4. Java How to Program, Deitel and Deitel, Pearson, 11th edition, 2017.