



SWAMINARAYAN UNIVERSITY

Program Name: Engineering

Level: Degree

Course/ Subject Code: BE103401

Course/Subject Name: Discrete Mathematics

w.e.f.Academic Year:	2023-24
Semester:	4 th Semester/ 2 nd Year
Category of the Course:	Basic science

Prerequisite:	Algebra , Logic
Rationale:	This course introduces the basic concepts of discrete mathematics in the field of computer science. It covers sets, logic, functions, relations, graph theory and algebraic structures. These basic concepts of sets, logic functions and graph theory are applied to Boolean Algebra and logic networks, while the advanced concepts of functions and algebraic structures are applied to finite state machines and coding theory.

Teaching and Examination Scheme -

Hours per Week			Course Credits	Examination Marks				100
L	T	P		Theory Marks		Practical Marks		
				E	M	I	V	
3	2	0	5	60	40	0	0	

Course Content:

Unit	Content	Hours	% of Weightage
1	Set Theory, Functions, and Counting: Basic Concepts of Set Theory (Definitions, Inclusion, Equality, Cartesian Product, Power Set, Operations, Venn Diagrams, Set Identities), Functions (Definition, Co-domain, Range, Image, Injective, Surjective, Bijective, Composition, Inverse, Identity, Properties), Counting Principles (Basics, Pigeonhole Principle, Permutations & Combinations, Binomial Coefficients, Generalized Forms, Generating Permutations & Combinations)	6	13%
2	Propositional and Predicate Logic: Statements, Truth Values, Connectives, Truth Tables, Well-formed Formulas, Tautologies, Equivalence, Duality, Implications; Predicate Logic: Predicates, Quantifiers, Free & Bound Variables, Universe of Discourse, Valid Formulas & Equivalences	6	13%
3	Relations, Partial Orderings, Lattices, and Recurrence Relations: Relations (Binary, Properties, Matrix, Graph, Equivalence Classes, Compatibility, Closure), Partial Ordering, Posets, Hasse Diagrams, LUB, GLB, Well-ordered Sets, Lattices (Basic Types), Recurrence Relations and Solving	10	22%
4	Algebraic Structures: One Binary Operation – Semigroup, Monoid, Group, Subgroup, Cosets, Homomorphism, Lagrange's Theorem, Quotient Structures; Two Binary Operations – Ring, Integral Domain, Field (Definitions and examples only)	10	22%



SWAMINARAYAN UNIVERSITY

Program Name: Engineering

Level: Degree

Course/ Subject Code: BE103401

Course/Subject Name: Discrete Mathematics

5	Graph Theory: Introduction, Graph Types (Directed, Undirected, Multigraph, Weighted), Degrees, Subgraphs, Paths, Cycles, Reachability, Connectedness, Components, Applications (Deadlock Detection), Matrix Representation (Adjacency, Boolean), Path Counting	10	22%
Total		45	100%

Suggested Specification Table with Marks (Theory):

Distribution of Theory Marks					
R Level	U Level	A Level	N Level	E Level	C Level
10	20	20	10	10	

Legends: R: Remembrance; U: Understanding; A: Application, N: Analyze and E: Evaluate C: Create and above Levels (Revised Bloom's Taxonomy).

Course Outcome:

After Completion of the Course, Student will able to:

Sr. No.	Course Outcome (CO)	RBT Level
CO1	Understand the basic principles of sets and operations in sets and apply counting principles to determine probabilities, domain and range of a function, identify one-to-one functions, perform the composition of functions and apply the properties of functions to application problems.	Understand, Apply
CO2	Write an argument using logical notation and determine if the argument is or is not valid. Simplify and evaluate basic logic statements including compound statements, implications, inverses, converses, and contrapositives using truth tables and properties of logic. Express logical sentences in terms of predicates, quantifiers, and logical connectives.	Understand, Apply, Analyze
CO3	Apply relations and determine their properties. Be familiar with recurrence relations.	Apply, Understand
CO4	Use the properties of algebraic structures.	Understand, Apply
CO5	Interpret different traversal methods for trees and graphs. Model problems in Computer Science using graphs and trees.	Understand, Apply, Analyze



SWAMINARAYAN UNIVERSITY

Program Name: Engineering

Level: Degree

Course/ Subject Code: BE103401

Course/Subject Name: Discrete Mathematics

CO-PO Mapping					
PO \ CO	CO1	CO2	CO3	CO4	CO5
PO1 – Engineering Knowledge	3	3	3	3	3
PO2 – Problem Analysis	2	3	3	2	3
PO3 – Design/Development	1	1	2	2	3
PO4 – Investigation	1	2	2	2	2
PO5 – Modern Tool Usage	2	2	1	1	3
PO6 – Engineer & Society	0	0	0	1	1
PO7 – Environment & Sustainability	0	0	0	0	1
PO8 – Ethics	0	1	1	1	1
PO9 – Individual & Team Work	0	1	1	1	1
PO10 – Communication	0	1	1	1	1
PO11 – Project Management	0	0	0	0	1
PO12 – Life-long Learning	0	1	1	1	3
CO-PSO Mapping					
PSO \ CO	CO1	CO2	CO3	CO4	CO5
PSO1 – Apply mathematical concepts to solve real-world computer science problems	3	3	3	3	3
PSO2 – Develop models and use discrete structures to analyze and design software systems	2	2	2	2	3

References/Suggested Learning Resources:

(a) Books:

- J. P. Tremblay and R. Manohar, Discrete Mathematical Structures with Applications to Computer Science, Tata McGraw-Hill, 1997.
- S. Lipschutz and M. L. Lipson, Schaum's Outline of Theory and Problems of Discrete Mathematics, 2nd Ed., Tata McGraw-Hill, 1999.
- K. H. Rosen, Discrete Mathematics and its applications, Tata McGraw-Hill, 6th Ed., 2007.
- David Liben-Nowell, Discrete Mathematics for Computer Science, Wiley publication, July 2017.
- Eric Gossett, Discrete Mathematics with Proof, 2nd Edition



SWAMINARAYAN UNIVERSITY

Program Name: Engineering

Level: UG

Course / Subject Code: BE103404

Course / Subject Name: Object Oriented Programming-1

w. e. f. Academic Year:	2023-24
Semester:	4 th
Category of the Course:	Professional Core

Prerequisite:	To get a solid grasp of Object-Oriented Programming (OOP), it helps to come in with a few building blocks already in place
Rationale:	Object oriented Programming has become a fundamental part of software development. OOP facilitates Reuse of code, flexibility, effective problem solving. It provides a modular structure for programs and implementation details are hidden. Reuse of code lowers the cost of development.

Course Outcome:

After Completion of the Course, Student will able to:

No	Course Outcomes	RBT level
1.	Use various Java constructs, features and libraries for simple problems.	Understand
2	Demonstrate how to define and use classes, interfaces, create objects and methods, how to override and overload methods, compile and execute programs.	Apply
3	Write a program using exception handling, multithreading with synchronization.	Apply
4.	Write a program using Files, binary I/O, collection Frameworks for a given problem.	Apply
5	Design and develop GUI based applications in a group using modern tools and frameworks.	Apply

CO PO PSO Mapping													
Course Outcomes (COs)	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2
CO1: Use various Java constructs, features, and libraries for simple problems.	3	2	2	-	2	-	-	-	-	2	-	3	2
CO2: Demonstrate how to define and use classes, interfaces, create objects and methods, override/overload.	3	2	2	-	2	-	-	-	-	2	-	3	2
CO3: Write a program using exception handling, multithreading with synchronization.	3	3	2	-	3	-	-	-	-	2	-	3	2

CO PO PSO Mapping													
Course Outcomes (COs)	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2
CO4: Write a program using Files, binary I/O, and collection frameworks for a given problem.	3	3	2	-	3	-	-	-	-	2	-	3	2
CO5: Design and develop GUI-based applications in a group using modern tools and frameworks.	2	2	3	2	3	-	-	-	2	2	2	3	3

Teaching and Examination Scheme:

Teaching Scheme (in Hours)			Total Credits L+T+ (PR/2)	Assessment Pattern and Marks				Total Marks
L	T	PR	C	Theory		Tutorial / Practical		
				(E)	(M)	(I)	(V)	
4	0	2	5	60	40	20	30	150



SWAMINARAYAN UNIVERSITY

Program Name: Engineering

Level: UG

Course / Subject Code: BE103404

Course / Subject Name: Object Oriented Programming-1

Course Content:

Unit No.	Content	No. of Hours	% of Weightage
1.	Introduction to java and elementary programming: Java language specification API, JDK and IDE, Creating, compiling and Executing a simple java program, Programming style, documentation and errors, Reading input from console, identifiers and variables, Assignment statements, Named constants and naming conventions, Data Types (Numeric, Boolean, Character, String) its Operations and Literals, Evaluating Expressions and operator Precedence, Types of Operators (Augmented assignment, Increment and Decrement, Logical), operator precedence and associativity, numeric type conversions.	4	10
2.	Selections, Mathematical functions and loops: If statements, Two way, Nested if and multi-way if statements, Switch statements, Conditional Expressions, Common mathematical functions, While, do-while and for loop, nested loops, Keyword break and continue.	4	10
3.	Methods and Arrays: Defining and calling method, Passing argument by values, Overloading methods and scope of variables, Method abstraction and stepwise refinement, Single Dimensional arrays, copying arrays ,Passing and returning array from method, Searching and sorting arrays and the Array class, Two-Dimensional array and its processing, Passing Two-dimensional Array to methods, Multidimensional Arrays.	6	10
4.	Objects and Classes: Defining classes for objects, Constructors, accessing objects via reference variable, using classes from the java library, static variables, constants and methods, visibility modifiers and Data field encapsulation, passing objects to methods, array of objects, immutable objects and classes, scope of variable and the this reference.	4	10
5.	Object oriented thinking: Class abstraction and Encapsulation, thinking in objects and class relationships, Primitive data type and wrapper class types, Big integer and Big decimal class, string class, String Builder and String Buffer class, super class and subclass, using super keyword, overriding and overloading methods, polymorphism and dynamic binding, casting objects and instance of operator, The ArrayList class and its methods, The protected data and methods.	5	5
6.	Exception Handling, I/O, abstract classes and interfaces: Exception types, finally clause, rethrowing Exceptions, chained exceptions, defining custom exception classes, file class and its input and output, Reading data from web, Abstract classes, interfaces, Comparable and Cloneable interface.	4	5



SWAMINARAYAN UNIVERSITY

Program Name: Engineering

Level: UG

Course / Subject Code: BE103404

Course / Subject Name: Object Oriented Programming-1

7.	JAVAFX basics and Event-driven programming and animations: Basic structure of JAVAFX program, Panes, UI control and shapes, Property binding, the Color and the Font class, the Image and Image View class, layout panes and shapes, Events and Events sources, Registering Handlers and Handling Events, Inner classes, anonymous inner class handlers, mouse and key events, listeners for observable objects, animation	5	10
8.	JAVAFX UI controls and multimedia: Labeled and Label, button, Checkbox, Radio Button, Text field, Text Area, Combo Box, List View, Scrollbar, Slider, Video and Audio.	4	5
9.	Binary I/O , Recursion and Generics: Text I/O, binary I/O, Binary I/O classes, Object I/o, Random Access files, Problem solving using Recursion, Recursive Helper methods, Tail Recursion, Defining Generic classes and interfaces, Generic methods, Raw types and backward compatibility, wildcard Generic types, Erasure and Restrictions on Generics.	4	10
10.	List, Stacks, Queues and Priority Queues: Collection, Iterators, Lists, The Comparator interface, static methods for list and collections, Vector and Stack classes, Queues and priority Queues.	4	10
11.	Sets and Maps: Comparing the performance of Sets and Lists, singleton and unmodifiable collections and Maps.	2	5
12.	Concurrency Thread states and life cycle, Creating and Executing threads with the Executor Framework, Thread synchronization	2	10
Total		48	100

Suggested Specification Table with Marks (Theory):

Distribution of Theory Marks					
R Level	U Level	A Level	N Level	E Level	C Level
20	40	40	-	-	-

Where R: Remember; U: Understanding; A: Application, N: Analyze and E: Evaluate C: Create (as per Revised Bloom's Taxonomy)

References/Suggested Learning Resources:

(a) Books:

1. Intro to Java Programming, 10th edition, Y.Daniel Liang, Pearson
2. Object oriented programming with Java , Rajkumar Buyya,S Thamarai Selvi, Xingchen Chu, McGraw Hill
3. Programming in Java, Sachin Malhotra, Saurabh Choudhary, Oxford
4. Programming with JAVA , E Balagurusamy, McGraw Hill
5. CORE JAVA volume -I Cay Horstmann, Pearson

(b) Open source software and website:

1. <https://nptel.ac.in>
2. www.vlab.co.in

Suggested Course Practical List:

1. Write a Program that displays Welcome to Java, Learning Java Now and Programming is fun.
2. Write a program that solves the following equation and displays the value x and y:
 - a. $3.4x + 50.2y = 44.5$
 - b. $2.1x + .55y = 5.9$ (Assume Cramer's rule to solve equation $ax + by = e$, $x = \frac{ed - bf}{ad - bc}$, $cx + dy = f$, $y = \frac{af - ec}{ad - bc}$)
3. Write a program that reads a number in meters, converts it to feet, and displays the result.
4. Body Mass Index (BMI) is a measure of health on weight. It can be calculated by taking your weight in kilograms and dividing by the square of your height in meters. Write a program that prompts the user to enter a weight in pounds and height in inches and displays the BMI.
 - a. Note :- 1 pound = .45359237 Kg and 1 inch = .0254 meters.
5. Write a program that prompts the user to enter three integers and display the integers in decreasing order.
6. Write a program that prompts the user to enter a letter and check whether a letter is a vowel or constant.
7. Assume a vehicle plate number consists of three uppercase letters followed by four digits. Write a program to generate a plate number.
8. Write a program that reads an integer and displays all its smallest factors in increasing order. For example if input number is 120, the output should be as follows: 2, 2, 2, 3, 5. Write a method with following method header. `public static int gcd (int num1, int num2)` Write a program that prompts the user to enter two integers and compute the gcd of two integers.
9. Write a test program that prompts the user to enter ten numbers, invoke a method to reverse the numbers, display the numbers.
10. Write a program that generate 6*6 two-dimensional matrix, filled with 0's and 1's, display the matrix, check every row and column have an odd number's of 1's.
11. Write a program that creates a Random object with seed 1000 and displays the first 100 random integers between 1 and 49 using the NextInt (49) method.
12. Write a program for calculator to accept an expression as a string in which the operands and operator are separated by zero or more spaces. For ex: 3+4 and 3 + 4 are acceptable expressions.
13. Write a program that creates an Array List and adds a Loan object, a Date object, a string, and a Circle object to the list, and use a loop to display all elements in the list by invoking the object's to String() method.
14. Write the bin2Dec (string binary String) method to convert a binary string into a decimal number. Implement the bin2Dec method to throw a Number Format Exception if the string is not a binary string.
15. Write a program that prompts the user to enter a decimal number and displays the number in a fraction. Hint: Read the decimal number as a string, extract the integer part and fractional part from the string.
16. Write a program that displays a tic-tac-toe board. A cell may be X, O, or empty. What to display at each cell is randomly decided. The X and O are images in the files X.gif and O.gif.
17. Write a program that moves a circle up, down, left or right using arrow keys.
18. Write a program that displays the color of a circle as red when the mouse button is pressed and as blue when the mouse button is released.
19. Write a GUI program that use button to move the message to the left and right and use the radio button to change the color for the message displayed.



SWAMINARAYAN UNIVERSITY

Program Name: Engineering

Level: UG

Course / Subject Code : BE103404

20. Write a program to create a file name 123.txt, if it does not exist. Append a new data to it if it already exist. write 150 integers created randomly into the file using Text I/O. Integers are separated by space.
21. Write a recursive method that returns the smallest integer in an array. Write a test program that prompts the user to enter an integer and display its product.
22. Write a generic method that returns the minimum elements in a two dimensional array.
23. Define MY Priority Queue class that extends Priority Queue to implement the Cloneable interface and implement the clone() method to clone a priority queue.
24. Write a program that reads words from a text file and displays all the nonduplicate words in descending order. The text file is passed as a command-line argument.

List of Laboratory/Learning Resources Required:

1. **OOP-Supporting Language:** Such as Java, C++, Python, C#, or Ruby.
2. **IDE or Code Editor:** Tools like IntelliJ IDEA, Eclipse, Visual Studio, or VS Code help manage OOP projects efficiently.
3. **Compiler/Interpreter:** Depending on the language, you'll need the appropriate runtime environment (e.g., JDK for Java, Python interpreter).

Suggested Project List:

1. Create a class `Student` with methods to compute and display grades.
2. Implement operator overloading for complex number arithmetic.
3. Demonstrate runtime polymorphism using virtual functions.
4. Build a mini project like a library management system using OOP principles.



सर्व विद्या अधिपतये

SWAMINARAYAN UNIVERSITY

Program Name: Engineering

Level: UG

Course / Subject Code: BE103405

Course / Subject Name: Computer Organization & Architecture

w. e. f. Academic Year:	2023-24
Semester:	4 th
Category of the Course:	Professional Core

Prerequisite:	To truly grasp Computer Organization and Architecture (COA), having a solid foundation in a few key areas can make the subject much more approachable and meaningful.
Rationale:	<ul style="list-style-type: none"> • Write better, optimized code • Choose the right hardware for specific applications • Design future-ready systems with emerging tech like quantum or neuromorphic computing in mind.

Course Outcome:

After Completion of the Course, Student will able to:

No	Course Outcomes	RBT level
1	Identify and explain the basic structure and functional units of a digital computer.	Understand
2	Write assembly language programs and identify the role and working of various functional units of a computer for executing an instruction.	Apply
3	Design processing unit using the concepts of ALU and control logic design.	Apply
4	Design circuits for interfacing memory and I/O with processor.	Apply
5	Comprehend the features and performance parameters of different types of computer architectures.	Understand

CO PO Mapping						
Course Outcomes (COs)	PO1 (Engineering Knowledge)	PO2 (Problem Analysis)	PO3 (Design/Development of Solutions)	PO4 (Investigations)	PO5 (Modern Tool Usage)	PO6–PO12 (Ethics, Society, Environment, etc.)
CO1: Identify and explain the basic structure and functional units of a digital computer.	3	2	1	1	1	—
CO2: Write assembly language programs and identify the role and working of various functional units of a	3	2	2	1	2	—

CO PO Mapping						
Course Outcomes (COs)	PO1 (Engineering Knowledge)	PO2 (Problem Analysis)	PO3 (Design/Development of Solutions)	PO4 (Investigations)	PO5 (Modern Tool Usage)	PO6–PO12 (Ethics, Society, Environment, etc.)
computer for executing an instruction.						
CO3: Design processing unit using the concepts of ALU and control logic design.	3	3	3	2	2	—
CO4: Design circuits for interfacing memory and I/O with processor.	3	2	3	2	2	—
CO5: Comprehend the features and performance parameters of different types of computer architectures.	3	2	2	2	1	—

CO PSO MAPPING			
CO \ PSO	PSO1	PSO2	PSO3
CO1	3	2	1
CO2	2	3	2
CO3	3	3	2
CO4	2	3	2
CO5	3	2	1

Teaching and Examination Scheme:

Teaching Scheme (in Hours)			Total Credits L+T+ (PR/2)	Assessment Pattern and Marks				Total Marks
L	T	PR	C	Theory		Tutorial / Practical		
				(E)	(M)	(I)	(V)	
3	2	0	5	60	40	20	30	150



SWAMINARAYAN UNIVERSITY

Program Name: Engineering

Level: UG

Course / Subject Code: BE103405

Course / Subject Name: Computer Organization & Architecture

Course Content:

Unit No.	Content	No. of Hours	% of Weightage
1.	Computer Data Representation: Basic computer data types, Complements, Fixed point representation, Register Transfer and Microoperations: Floating point representation, Register Transfer language, Register Transfer, Bus and Memory Transfers (Tree-State Bus Buffers, Memory Transfer), Arithmetic Micro Operations, Logic Micro-Operations, Shift Micro-Operations, Arithmetic logical shift unit .	4	10
2.	Basic Computer Organization and Design: Instruction codes, Computer registers, computer instructions, Timing and Control, Instruction cycle, Memory-Reference Instructions, Input-output and interrupt, Complete computer description, Design of Basic computer, Design of Accumulator Unit.	4	10
3.	Assembly Language Programming: Introduction, Machine Language, Assembly Language Programming: Arithmetic and logic operations, looping constructs, Subroutines, I-O Programming.	8	10
4.	Micro programmed Control Organization: Control Memory, Address sequencing, Micro program example, Design of Control Unit	4	10
5.	Central Processing: Unit Introduction, General Register Organization, Stack Organization, Instruction format, Addressing Modes, Data transfer and manipulation, Program control, Reduced Instruction Set Computer (RISC) & Complex Instruction Set Computer (CISC)	5	10
6.	Pipeline and Vector Processing: Flynn's taxonomy, Parallel Processing, Pipelining, Arithmetic Pipeline, Instruction, Pipeline, RISC Pipeline, Vector Processing, Array Processors	5	10
7	Computer Arithmetic: Introduction, Addition and subtraction, Multiplication Algorithms (Booth Multiplication Algorithm), Division Algorithms, Floating Point Arithmetic operations, Decimal Arithmetic Unit.	4	10
8	Input-Output Organization: Input-Output Interface, Asynchronous Data Transfer, Modes of Transfer, Priority Interrupt, DMA, Input-Output Processor (IOP), CPU IOP Communication, Serial communication.	4	10



SWAMINARAYAN UNIVERSITY

Program Name: Engineering

Level: UG

Course / Subject Code: BE103405

Course / Subject Name: Computer Organization & Architecture

9.	Memory Organization: Memory Hierarchy, Main Memory, Auxiliary Memory, Associative Memory, Cache Memory, Virtual Memory.	6	10
10.	Multiprocessors: Characteristics of Multiprocessors, Interconnection Structures, Inter-processor Arbitration, Inter-processor Communication and Synchronization, Cache Coherence, Shared Memory Multiprocessors.	4	10
Total		48	100

Suggested Specification Table with Marks (Theory):

Distribution of Theory Marks					
R Level	U Level	A Level	N Level	E Level	C Level
20	40	40	-	-	-

Where R: Remember; U: Understanding; A: Application, N: Analyze and E: Evaluate C: Create (as per Revised Bloom's Taxonomy)

References/Suggested Learning Resources:

(a) Books:

1. M. Morris Mano, "Computer System Architecture", Pearson Education
2. Yale N. Patt, Sanjay J. Patel, "Introduction to Computing Systems" McGraw Hill.
3. Hamacher, Vranesic, Zaky, "Computer Organization", McGraw Hill.
4. Andrew S. Tanenbaum and Todd Austin, "Structured Computer Organization", Pearson Education
5. N. D. Jotwani, "Computer system organization", McGraw Hill
6. R.S.Gaonkar, "Microprocessor Architecture, Programming and Applications with 8085A", Penram International
7. Douglas Hall, Microprocessors and Interfacing, TMH.

(b) Open source software and website:

- 1.

Suggested Course Practical List:

1. Implement Booth's Algorithm
2. Write the working of 8085 simulator GNUsim8085 and basic architecture of 8085 along with small introduction.
3. Write an assembly language code in GNUsim8085 to store numbers in reverse order in memory location.
4. Write an assembly language code in GNUsim8085 to implement arithmetic instruction.



SWAMINARAYAN UNIVERSITY

Program Name: Engineering

Level: UG

Course / Subject Code: BE103405

Course / Subject Name: Computer Organization & Architecture

5. Write an assembly language code in GNUsim8085 to find the factorial of a number.
6. Write an assembly language code in GNUsim8085 to implement logical instructions.
7. Design ALU using Logisim.
8. Implement 16-bit single-cycle MIPS processor in Verilog HDL

List of Laboratory/Learning Resources Required:

1. Identification and Explanation of Computer Components
2. Assembly Language Programming
3. Design of Arithmetic Logic Unit (ALU)
4. Control Unit Design
5. Memory and I/O Interfacing
6. Comparative Study of Computer Architectures
7. Pipelining and Hazards
8. Cache Memory Organization
9. Interrupts and DMA

Suggested Project List:

1. Write assembly language programs and identify the role and working of various functional units of a computer for executing an instruction
2. Design processing unit using the concepts of ALU and control logic design
3. Design circuits for interfacing memory and I/O with processor
4. Comprehend the features and performance parameters of different types of computer architectures

* * * * *



SWAMINARAYAN UNIVERSITY

Program Name: Engineering

Level: UG

Course / Subject Code: BE104401

Course / Subject Name: Operating System and Virtualization

w. e. f. Academic Year:	2023-24
Semester:	4 th
Category of the Course:	Professional Core

Prerequisite:	Linear and non-linear data structures, working experience of any one structured programming language
Rationale:	process management, memory allocation, file systems, and synchronization. Learning to write shell scripts, simulate scheduling algorithms, and explore OS internals. Prepares students for fields like systems programming, cybersecurity, and distributed computing.

Course Outcome:

After Completion of the Course, Student will able to:

No	Course Outcomes	RBT level
1	Analyze the structure of OS and basic architectural components involved in OS design	Understand
2	Compare and contrast various CPU scheduling algorithms.	Apply
3	Evaluate the requirements for the process synchronization and co-ordination in contemporary operating system.	Apply
4	Analyze various algorithms for memory management, I/O management and security aspects of operating system.	Understand
5	Write shell scripts in Unix/Linux O.S and write simple programs using kernel system calls. Also understand virtualization concept.	Understand

CO PO Mapping												
CO \ PO	PO1(Engg. Knowledge)	PO2(Problem Analysis)	PO3(Design/Dev)	PO4(Mode rn Tools)	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12(Life-long Learning)
CO1	3	2	1	1	-	-	-	-	-	-	-	2
CO2	3	3	2	1	-	-	-	-	-	-	-	2
CO3	3	3	2	1	-	-	-	-	-	-	-	2
CO4	3	3	2	2	-	-	-	-	-	-	-	2
CO5	3	2	2	3	-	-	-	-	1	2	1	3

CO PSO Mapping		
Course Outcome (CO)	Description	Mapped PSOs
CO1: Analyze the structure of Operating System and basic architectural components involved in OS design.	Understand OS functions and core design principles.	PSO1, PSO2
CO2: Compare and contrast various CPU scheduling algorithms.	Evaluate performance of different scheduling techniques (FCFS, SJF, RR, Priority, etc.).	PSO1, PSO2
CO3: Evaluate the requirements for process synchronization and co-ordination in contemporary OS.	Understand synchronization primitives, deadlocks, semaphores, monitors, etc.	PSO1, PSO2
CO4: Analyze various algorithms for memory management, I/O management and security aspects of OS.	Evaluate paging, segmentation, virtual memory, disk scheduling, and security mechanisms.	PSO1, PSO2
CO5: Write shell scripts in Unix/Linux OS and write simple programs using kernel system calls. Understand virtualization.	Demonstrate scripting and system-level programming skills. Understand basics of virtual machines and containers.	PSO2, PSO3

Teaching and Examination Scheme:

Teaching Scheme (in Hours)			Total Credits L+T+ (PR/2)	Assessment Pattern and Marks				Total Marks
L	T	PR	C	Theory		Tutorial / Practical		
				(E)	(M)	(I)	(V)	
4	0	2	5	60	40	20	30	150



SWAMINARAYAN UNIVERSITY

Program Name: Engineering

Level: UG

Course / Subject Code: BE104401

Course / Subject Name: Operating System and Virtualization

Course Content:

Unit No.	Content	No. of Hours	% of Weightage
1.	Introduction: Computer system overview, Architecture, Goals & Structures of O.S, Basic functions, Interaction of O.S. & hardware architecture, System calls, Batch, multiprogramming. Multitasking, time sharing, parallel, distributed & realtime O.S.	5	10
2.	Process and Threads Management: Process Concept, Process states, Process control, Threads, Uni-processor Scheduling: Types of scheduling: Preemptive, Non preemptive, Scheduling algorithms: FCFS, SJF, RR, Priority, Thread Scheduling, Real Time Scheduling. System calls like ps, fork, join, exec family, wait.	8	10
3.	Concurrency: Principles of Concurrency, Mutual Exclusion: S/W approaches, H/W Support, Semaphores, Pipes, Message Passing, Signals, Monitors.	4	5
4.	Inter Process Communication: Race Conditions, Critical Section, Mutual Exclusion, Hardware Solution, Strict Alternation, Peterson's Solution, The Producer Consumer Problem, Semaphores, Event Counters, Monitors, Message Passing, Classical IPC Problems: Reader's & Writer Problem, Dining Philosopher Problem etc., Scheduling, Scheduling Algorithms	8	15
5.	Deadlock: Principles of Deadlock, Starvation, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, System calls.	4	10
6.	Memory Management: Memory Management requirements, Memory partitioning: Fixed and Variable Partitioning, Memory Allocation: Allocation Strategies (First Fit, Best Fit, and Worst Fit), Swapping, Paging and Fragmentation. Demand Paging, Security Issues. Virtual Memory: Concepts, VM management, Page Replacement Policies (FIFO, LRU, Optimal, Other Strategies), Thrashing	8	15
7	I/O Management & Disk scheduling: I/O Devices, Organization of I/O functions, Operating System Design issues, I/O Buffering, Disk Scheduling (FCFS, SCAN, C-SCAN, SSTF), RAID, Disk Cache.	6	10



SWAMINARAYAN UNIVERSITY

Program Name: Engineering

Level: UG

Course / Subject Code: BE104401

Course / Subject Name: Operating System and Virtualization

8.	Security & Protection: Security Environment, Design Principles of Security, User Authentication, Protection Mechanism: Protection Domain, Access Control List	4	10
9.	Unix/Linux Operating System: Development of Unix/Linux, Role & Function Of Kernel, System Calls, Elementary Linux command & Shell Programming, Directory Structure, System Administration Case study: Linux, Windows Operating System	4	10
10	Virtualization Concepts: Virtual machines; supporting multiple operating systems simultaneously on a single hardware platform; running one operating system on top of another. True or pure virtualization	3	5
Total		54	100

Note: Topic No. 4 and 7 of the above syllabus must be covered in Practical Hours.

Suggested Specification Table with Marks (Theory):

Distribution of Theory Marks					
R Level	U Level	A Level	N Level	E Level	C Level
20	40	40	-	-	-

Where R: Remember; U: Understanding; A: Application, N: Analyze and E: Evaluate C: Create (as per Revised Bloom's Taxonomy)

References/Suggested Learning Resources:

(a) Books:

1. Operating Systems: Internals & Design Principles, 9 th Edition, William Stallings, Pearson Education India
2. Operating System Concepts, 9th edition Peter B. Galvin, Greg Gagne, Abraham Silberschatz, John Wiley & Sons, Inc.
3. Modern Operating Systems-By Andrew S. Tanenbaum (PHI)

(b) Open source software and website:

1. <https://ubuntu.com/>
2. <https://getfedora.org/>
3. <https://debian.org/>
4. <https://freebsd.org/>



SWAMINARAYAN UNIVERSITY

Program Name: Engineering

Level: UG

Course / Subject Code: BE104401

Course / Subject Name: Operating System and Virtualization

List of Laboratory/Learning Resources Required:

1. Writing shell scripts
2. Simulating CPU scheduling algorithms
3. Implementing memory allocation strategies
4. Creating multithreaded programs
5. File handling and system call experiments

Suggested Project List:

1. Analyze the Structure of OS and Architectural Components
2. Compare and Contrast Various CPU Scheduling Algorithms
3. Process Synchronization and Coordination
4. Memory Management, I/O, and Security Analysis
5. Shell Scripting and Kernel System Calls
6. Understanding Virtualization Concepts

* * * * *