B.N.M. Institute of Technology

An Autonomous Institution under VTU, Approved by AICTE

Department of Mathematics

Syllabus

Semester: III			
Course: Fourier Transform, Mathematical logic and Advanced Linear Algebra			
Course Code: 24MAC131 (Common to CSE, ISE, AIML)			
L:T:P:J	**2:2:0:0**	**CIA**	**: 50**
Credits:	**03**	**SEA**	**: 50**
Hours:	**40**	**SEA Duration**	**: 03 Hours**

Course Learning Objectives: The students will be able to

1 Have an insight into Fourier series, Fourier transforms.

2 Develop knowledge of Fundamentals of logic and Relations, Vector Spaces, Linear Transformation & Inner product spaces arising in engineering

Module-1: Fourier Series & Fourier Transforms	No. of hours	Blooms cognitive Levels
Examples from Engineering that require Fourier series and Fourier Transforms. **Fourier series**: Periodic functions, Introduction to Fourier Series, Dirichlet's condition. Problems on Fourier series over $(-l, l)$. **Fourier Transforms:** Introduction to infinite Fourier transform, Fourier sine and cosine transform and properties, problems on infinite Fourier transform, Discrete & Fast Fourier transform. *Experiential Learning component: Finding the Fourier series and Fourier Transform of a function*	**L : 04** **T : 04**	**L1** **L2** **L3**
Module-2: Mathematical logic and Boolean Algebra		
Examples from Engineering that require Fundamentals of logic and Relations. **Mathematical logic:** Basic connectives and truth tables, logic equivalence - the laws of logic, logical implication- rules of inference **Boolean Algebra:** Boolean functions, Representation of Boolean functions, Logic gates, minimization of circuits. *Experiential Learning component: Construction of combinational and sequential circuit.*	**L : 04** **T : 04**	**L1** **L2** **L3**
Module-3: Vector Spaces		
Examples from Engineering that require vector spaces Recap of system of linear homogenous and non-homogeneous equation and solution sets. Vector spaces, subspaces, linearly independent and dependent, Linear span of a set, Basis and dimension, coordinate vectors. *Experiential Learning component: Problems on linearly independent and dependent vectors, basis and dimension of a vector space.*	**L : 04** **T : 04**	**L1** **L2** **L3**
Module-4:Linear Transformation		
Examples from Engineering that require linear transformation. Linear transformations, algebra of linear transformations, representation of transformations by matrices, Non-singular linear transformation, Inverse of a linear transformation, Range space, Null space and problems on Rank-nullity theorem. *Experiential Learning component: Problems on Inverse of a linear transformation and Rank-nullity theorem*	**L : 04** **T : 04**	**L1** **L2** **L3**
Module-5: Inner Product Spaces		
Examples from Engineering that require Inner product spaces. Inner products Inner product spaces, Orthogonal set, orthogonal projections, orthonormal bases, Gram-Schmidt process, QR-factorization, Recap of Eigen values and Eigen vectors, problems on Singular value decomposition. *Experiential Learning component: Problems on QR-factorization and singular value decomposition*	**L : 04** **T : 04**	**L1** **L2** **L3**

Course Outcomes: After completing the course, the students will be able to	
CO 1:	Apply Fourier series & transform concepts in data visualization and cryptography.
CO 2:	Convert Boolean expressions to logic gates and vice-versa.
CO 3:	Apply the knowledge of vector spaces for solving problems in arising in engineering field
CO 4:	Apply the knowledge of linear transform for solving problems in arising in image processing
CO 5:	Compute orthogonal and orthonormal bases vectors and decomposition of a symmetric matrix using standard technique.

CO - PO Mapping:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO 1	3	2			2							
CO 2	3	2			2							
CO 3	3	2			2							
CO 4	3	2			2							
CO 5	3	2			2							

Reference Books:

1. E. Kreyszig: "Advanced Engineering Mathematics", John Wiley & Sons,10th Edition(Reprint), 2016.
2. B. S. Grewal: "Higher Engineering Mathematics", Khanna Publishers, 44th Ed., 2017.
3. C. Ray Wylie, Louis C.Barrett : "Advanced Engineering Mathematics", 6" Edition, 2. McGraw-Hill Book Co., New York, 1995.
4. James Stewart : "Calculus —Early Transcendentals", Cengage Learning India Private Ltd., 2017.
5. Srimanta Pal & Subodh C Bhunia: "Engineering Mathematics", Oxford University Press, 3"Reprint, 2016.
6. David C. Lay, Steven R. Lay and J. J. McDonald "Linear Algebra and its applications", 3rd Edition, Pearson Education Ltd., 2017.
7. Kenneth H Rosen, "Discrete Mathematics and its Applications, Special Indian Edition 2021, McGraw Hill publication (India).
8. Ralph P. Grimaldi, " Discrete and Combinatorial Mathematics, 5th Edition, Pearson Education 2004.

Web links and Video Lectures:

1. https://nptel.ac.in/courses/111106111
2. https://youtu.be/OynpZwylau8
3. https://archive.nptel.ac.in/courses/111/106/111106051/
4. https://www.youtube.com/watch?v=zvRdbPMEMUI
5. https://www.youtube.com/watch?v=PiG2BMkK3s4
6. https://www.youtube.com/watch?v=ATqV_I8DCh0

B.N.M. Institute of Technology

An Autonomous Institution under VTU
Department of Information Science and Engineering

Course Name: Computer Organization
Course Code: 24ISE132

SEMESTER–III

L:T:P:J:	3:0:0:0	**CIE Marks:**	50
Credits:	3	**SEA Marks:**	50
Total Number of Lecture Hours	40	**SEA Duration:**	3 Hours

Course objectives:

- To introduce the fundamental subsystems of a computer, including their organization, structure, and operational principles, with a focus on how machine instructions form executable programs.
- To develop a strong conceptual understanding of memory hierarchy and organization, including primary memory, cache memory, and their role in enhancing system performance.
- To equip students with the ability to perform arithmetic and logical operations in digital systems and to understand the functioning and significance of pipelining and multiprocessor architectures in improving computational efficiency.

Module1: Introduction	**No. of Hours**	**Blooms Cognitive Levels**
Basic Structure of Computers: Basic Operational Concepts, Bus Structures, Performance – Processor Clock, Basic Performance Equation, Clock Rate, Performance Measurement. Machine Instructions and Programs: Memory Location and Addresses, Memory Operations, Instructions and Instruction Sequencing, Addressing Modes, Assembly Language, Basic Input and Output Operations, **Stacks and Queues, Subroutines**, Additional Instructions, Encoding of Machine Instructions.	**8**	**Understand**
Module 2: Input / Output Organization		
Accessing I/O Devices, Interrupts – Interrupt Hardware, Direct Memory Access, Buses, Interface Circuits, **Standard I/O Interfaces – PCI Bus, USB.**	**8**	**Understand**
Module 3: Memory System		
Memory System: Basic Concepts, Semiconductor RAM Memories, Read Only Memories, Speed, Size, and Cost, Cache Memories – Mapping Functions, **Replacement Algorithms**, Performance Considerations.	**8**	**Apply**
Module 4: MIPS Arithmetic operations		
MIPS Addressing for 32-Bit Immediate and Addresses, Parallelism and Instructions: Synchronization, Translating and Starting a Program, Addition and Subtraction (MIPS), Multiplication and Division (MIPS).	**8**	**Apply**

Module 5: Pipelining and Multiprocessors	**No. of Hours**	**Blooms Cognitive Levels**
Pipelining: Basic concepts, Data Hazards, Instruction hazards, Basic processing unit: some fundamental concepts, execution of complete instruction, multi-bus organization, The structure of general-purposes multiprocessors, **Parallel Computer Architecture: Processor Architecture and Technology Trends, Flynn's Taxonomy of Parallel Architectures.**	**8**	**Analyze**

Course outcomes: The students should be able to:

- **CO1:** Understand the fundamental concepts of computer systems including instruction formats, memory hierarchy, input/output organization, MIPS instruction set, and pipelining in multiprocessors.
- **CO2:** Construct basic assembly language programs using MIPS, and design basic memory and I/O interfacing systems for effective data transfer.
- **CO3:** Analyze the performance of computer systems by evaluating instruction cycles, memory access techniques, pipelining efficiency, and multiprocessor architectures using standard metrics and models.

Reference Books:

1. Carl Hamacher, Zvonko Vranesic, Safwat Zaky, Naraig Manjikian, Computer Organization and Embedded Systems, 6th Edition (Reprint), McGraw Hill Education, 2017.
2. Thomas Rauber, Gudula Rünger, Parallel Programming for Multicore and Cluster Systems, 3rd Edition, Springer, 2023.
3. David A. Patterson, John L. Hennessy, Computer Organization and Design: The Hardware/Software Interface, 6th Edition, Morgan Kaufmann / Elsevier, 2021.
4. William Stallings, Computer Organization and Architecture, 11th Edition, Pearson Education, 2021.

B.N.M. Institute of Technology

An Autonomous Institution under VTU
Department of Information Science and Engineering

Operating Systems

Course Code: 24ISE133

SEMESTER–III

L:T:P:J:	3:0:0:0	**CIA Marks:**	50
Credits:	3	**SEA Marks:**	50
Total Number of Lecture Hours	40	**SEA Duration:**	3 Hours

Course Objectives:

- Understand and analyze the core principles of operating system design and functionality, including process management, CPU scheduling, memory management, and deadlock handling.
- Develop the ability to implement and evaluate synchronization mechanisms and memory management techniques, including page replacement and disk scheduling algorithms, through practical programming exercises.
- Gain proficiency in UNIX system usage, including file system navigation, command-line operations, file permission handling, and shell scripting, to effectively automate tasks and manage system resources.

Module 1: Introduction to Operating System & Process Management	**No. of Hours**	**Blooms Cognitive Levels**
Fundamental Concepts of Operating System: Introduction to Operating systems, Operating system functions and services, historical evolution of operating systems, System boot. **Process Management:** Process abstraction, process address space, process management, system calls, threads. **CPU Scheduling:** Levels of scheduling, comparative study of scheduling algorithms, Multilevel Queue Scheduling, Multilevel Feedback Queue Scheduling, Multi-processor scheduling. Practical Component: **CPU Scheduling Programs not limited to FCFS, SJF, Pre-emptive and non-Pre-emptive.**	**8**	**Analyze**
Module 2: Process Synchronization and Deadlocks		
Concurrent Processes: Critical section problem, semaphores, Classical problems of synchronization, monitors, inter-process communication, message passing mechanisms. Deadlocks: Characterization, prevention and avoidance, deadlock detection and recovery. Practical Component: **Synchronization Programs not limited to readers writers' problem, bounded buffer problem.**	**8**	**Analyze**
Module 3: Memory Management		
Memory Management: Background; Swapping; Contiguous memory allocation; Paging; Structure of page table; Segmentation, virtual memory concept, demand paging, page replacement algorithms, thrashing, Disk Scheduling. Solid State Drives-SSD Architecture, Flash Controller, Practical Component: **Disk Scheduling Programs not limited to FCFS, SCAN, LOOK.**	**8**	**Analyze**

Module 4: UNIX file system	No. of Hours	Blooms Cognitive Levels
Unix files: UNIX Architecture, Naming files. **Basic file types/categories. Organization of files.** Hidden files. Standard directories. Parent child relationship. The home directory and the HOME variable. Reaching required files- the PATH variable, manipulating the PATH, Relative and absolute pathnames. Directory commands – pwd, cd, mkdir, rmdir commands. The dot (.) and double dots (..) notations to represent present and parent directories and their usage in relative path names. File related commands – cat, mv, rm, cp, wc and od commands. **Practical component:** Execution of UNIX Shell Commands.	**8**	**Apply**
Module 5: File attributes and Shell programming		
File attributes and permissions: The ls command with options. Changing file permissions: the relative and absolute permissions changing methods. Recursively changing file permissions. Directory permissions. **The shells interpretive cycle:** Wild cards. Removing the special meanings of wild cards. Three standard files and redirection. Connecting commands: Pipe, grep, egrep. **Shell programming:** Ordinary and environment variables. Read and read-only commands. Command line arguments. Logical operators for conditional execution. The test command and its shortcut. The if, while, for and case control statements. The set and shift commands and handling positional parameters. The here (<<) document. Simple shell program examples. Practical component: **Execution of Wildcards & UNIX Shell Programs with respect to conditional statements.**	**8**	**Analyze**

Course Outcomes: After completing the course, the students will be able to	
CO 1	Understand the fundamental concepts of operating systems, process management, memory organization, and UNIX file system structures.
CO 2	Apply the concepts of shell scripts, CPU & disc scheduling, process synchronization, and UNIX file operations.
CO 3	Analyze and compare Shell wild cards, UNIX pattern matching commands, CPU scheduling & Disc Scheduling strategies, deadlock handling techniques, and memory management schemes to improve system performance.

Reference Books:

1. Sumitabha Das., Unix Concepts and Applications, 4th Edition, Tata McGraw Hill, November 8, 2017 ISBN: 978-0070635463.
2. Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, Operating System Principles 10th edition, Wiley-India, April 15, 2018.
3. William Stallings, Operating System: Internals and Design Principles, Prentice Hall, 9th Edition, 2018.
4. W. Richard Stevens: Advanced Programming in the UNIX Environment, 3rd Edition, Pearson Education, 2013.
5. Unix System Programming Using C++ - Terrence Chan, 1st Edition, PHI, 1996.

Web links and Video Lectures:

1. https://academicearth.org/
2. https://archive.nptel.ac.in/courses/106/105/106105214/

B.N.M. Institute of Technology

An Autonomous Institution under VTU
Department of Information Science and Engineering

Course Name: Data Structures and its Applications
Course Code: 24ISE134

SEMESTER– III

L:T:P:J:	3:0:2:0	**CIE Marks:**	50
Credits:	4	**SEA Marks:**	50
Total Number of Lecture Hours	50	**SEA Duration:**	3 Hours

Course objectives:

1. To introduce fundamental linear data structures such as stacks, queues, and linked lists, along with their operations, implementations, and practical applications in problem-solving.
2. To equip with the understanding and implementation of hierarchical and graph data structures, focusing on their properties, traversals, and real-world use cases.
3. To provide a solid understanding of advanced data structures including heap trees and hashing techniques, covering key operations like insertion, deletion, and collision handling.
4. To introduce efficient file organization and indexing techniques, including primary and secondary indexing, B-trees, and multi-way merging for optimized data access.

Module 1: Stacks and Queues	**No. of Hours**	**Blooms Cognitive Levels**
Data Structures: Introduction to Structures, Classification (Primitive & Non-primitive), Operations, Pattern Matching Algorithms (Brute force, KMP) Stacks: Definition, Operations, Implementation using arrays, Applications of Stacks – Infix to Postfix Conversion and Postfix Expression Evaluation. Queues: Definition, Operations, Implementation, Applications, **Circular Queue (Message queue using Circular queue), Doubly Ended Queue, Priority Queue**. **Practical Component:** Tracing Exercises, Analysis Exercises, Exercises to Modify programs. **Programs not limited to:** 1. Implement KMP pattern matching algorithm for a given main string and pattern. 2. Implement Boyer-Moore pattern matching algorithm for a given main string and pattern. 3. Program to implement stack operations (push, pop, and display) using static array and also on dynamic array and compare the performance. 4. Program to convert infix expression into postfix. 5. Program to evaluate given postfix expression. 6. Program to implement queue operations using static arrays and on dynamic arrays and compare the performance. 7. Program to implement circular queue operations using arrays and on dynamic arrays and compare the performance.	**10**	**Apply**

Module 2: Linked List	**No. of Hours**	**Blooms Cognitive Levels**
Linked Lists**:** Definition, Create, Insert, Delete, Update, Traverse, and Position-based Operations, Concatenate, Merge, and Reverse Lists, Doubly Linked List Implementation and Operations, Circular Linked List Implementation and Operations, **Applications of Lists (Polynomial addition)**. **Implementation of stacks and queues using Linked List.** **Practical Component:** Tracing Exercises, Analysis Exercises, Exercises to Modify programs. **Programs not limited to:** 1. Program to create single linked list and implement its operations with and without header nodes: i. Insert (front and rear end) ii. Delete. (Front and rear end) iii. Search. iv. Reverse. 2. Implement DLL as a stack and also as a queue. 3. Create 2 CLL of sparse polynomials and perform addition on these 2 polynomials. Represent the resulting polynomial in a CLL. 4. Implement multiple stacks and multiple queues in a singly linked list.	**10**	**Apply**
Module 3: Trees		
Concept Learning: Trees: General Tree Representation, Traversals, Applications. Binary Trees: Definition, Properties, Traversals, Applications. Binary Search Tree: Definition, Implementation, Search, Insert, Delete operations. Building and Evaluating Binary Expression Tree. **Concept Learning Graphs: Adjacency Lists, and Traversal of Graphs (BFS and DFS)** **Practical Component:** Tracing Exercises, Analysis Exercises, Exercises to Modify programs. **Programs not limited to:** 1. Implement the following operations on Binary tree: - A. Count the number of nodes in the binary tree B. Count the leaf nodes C. Check if 2 trees are equal D. Perform Tree Traversals namely in-order, pre-order, post-order and level order. 2. Create a Binary Search Tree for a set of integer values, perform deletion of a key and searching for a key. 3. For a given input graph check whether the nodes are connected or not using BFS and DFS method. 4. Create an expression tree to evaluate the given expression using binary tree.	**10**	**Apply**
Module 4: Advanced Trees & Hashing		
Concept Learning: Heap Tree: Definition, Implementation, Insert, Delete, Peek operations. **Hashing**: **Hash Table, Hash Functions, Collision Handling by Open Addressing, Chaining.**	**10**	**Apply**

Practical Component: Tracing Exercises, Analysis Exercises, Exercises to Modify programs. **Programs not limited to:** 1. Create a max heap and min heap of integers, display the values and perform deletion operations. 2. Program to implement hashing with collision resolution.		
Module 5: File Structures	**No. of Hours**	**Blooms Cognitive Levels**
File Structures**:** Primary Indexing, Secondary Indexing, B-Trees, Co sequential processing, **K way merge**. **Practical Component:** Tracing Exercises, Analysis Exercises, Exercises to Modify programs. **Programs not limited to:** 1. Write a program to implement simple index on primary key for a file. Implement add (), search (), delete () using the index. 2. Write a program to read two lists of names and then match the names in the two lists using Consequential Match based on a single loop. Output the names common to both the lists. 3. Write a program to read k Lists of names and merge them using k-way merge algorithm with k = 8.	**10**	**Apply**

Course outcomes: The students should be able to:

1. Understand the fundamental concepts of linear and non-linear data structures, their types, operations, and applications in problem-solving, along with file structures like indexing, B-trees, and k-way merge.
2. Apply various linear and non-linear data structures such as stacks, queues, linked lists, trees, graphs, and file structures to develop efficient solutions for real-world problems.
3. Apply static and dynamic memory techniques to implement linear and non-linear data structures such as stacks, queues, linked lists, trees, graphs, heaps, and file structures effectively to solve computational problems.
4. Apply pattern matching, expression evaluation, traversals, hashing, and file operations to ensure correct and efficient solutions.

Reference Books:

1. Data Structures Using C – Reema Thareja, 3rd Edition, Oxford Higher Education, 2023, ISBN-13: 978-9354977190.
2. Introduction to Algorithms – T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, 4th Edition, MIT Press, MA, 2022.
3. Data Structures and Algorithm Analysis in C++ – Mark Allen Weiss, 4th Revised Edition, Addison-Wesley, 2014, ISBN-13: 978-0132847377.

Weblinks:

1. https://youtube.com/playlist?list=PLBF3763AF2E1C572F&feature=shared
2. https://youtu.be/gJ6f3qGwGAU?feature=shared

B.N.M. Institute of Technology

An Autonomous Institution under VTU
Department of Information Science and Engineering

Course Code: Object Oriented Programming using Java
Course Code: 24ISE135

III Semester

L: T:P: J	3:0:2:0	**CIA Marks**	50
Credits	04	**SEA Marks**	50
Total Number of Contact Hours	50	**SEA Duration**	3 Hours

Course Learning Objectives: This course will enable students to:

- Show competence in the use of the Java programming language in the development of small to medium-sized application programs that demonstrate professionally acceptable coding and performance standard.
- Understand the basic principles of the object-oriented programming.
- Demonstrate an introductory understanding of graphical user interfaces, multithreaded programming, and event-driven programming.

Module – 1	**No. of Hours**	**Blooms Cognitive Levels**
Basics of Java: Features of Java, Byte Code and Java Virtual Machine, JDK, Data types, Operator, **Control Statements – If , else, nested if, if-else ladders, Switch, while, do-while, for, for-each, break, continue.** **Array :** Single and Multidimensional Array, **Classes, Objects and Methods:** Class, Object, Object reference, Constructor, Constructor Overloading, Method Overloading, Recursion, Passing and Returning object form Method, new operator, this and static keyword, finalize() method , Access control, modifiers, Nested class, Inner class, Anonymous inner class, Abstract class. **Programs:** 1. Write a program to convert rupees to dollar. 60 rupees=1 dollar. 2. Write a program that calculate the percentage marks of the student if marks of 6 subjects are given. 3. Write a program to enter two numbers and perform mathematical operations on them. 4. Write an interactive program to print a diamond shape. 5. Program to define class Box and constructors. Demonstrate constructors. 6. Program to define class Addition, methods and objects. Demonstrate method overloading to add 2 integers, 3 floating numbers and 2 strings.	**12**	**Apply**
Module – 2		
String: String class, StringBuffer class, Operations on string, Command line argument, Use of Wrapper Class. **Inheritance:** Use of Inheritance, Inheriting Data members and Methods, constructor in inheritance, Multilevel Inheritance – method overriding Handle multilevel constructors – super keyword, Stop Inheritance - Final keywords, Dynamic method dispatch, Understanding of Java Object Class. **Package: Use of Package, CLASSPATH, Import statement, Static import, Access control.**		

	No. of Hours	Blooms Cognitive Levels
Programs: 1. Write a program to find length of string and print second half of the string. 2. Write a program to accept a line and check how many consonants and vowels are there in line. 3. Write a program to count the number of words that start with capital letters. 4. Write a program to find that given number or string is palindrome or not. 5. Create a class which asks the user to enter a sentence, and it should display count of each vowel type in the sentence. The program should continue till user enters a word "quit". Display the total count of each vowel for all sentences. 6. Write an interactive program to print a string entered in a pyramid form. 7. Program to define inheritance and show method overriding - Program to define base class Shapes, with methods and create derived classes Square, Triangle, Rectangle and Circle. Create objects and show method overriding for calculating area of Square, Triangle, Rectangle and Circle.	**10**	**Apply**
Module – 3		
Interfaces : Creation and Implementation of an interface, Interface reference, Interface inheritance, Nesting Interfaces, variables in Interfaces, Comparison between Abstract Class and Interface. **Exception Handling:** Exception and Error, Use of try, catch, throw, throws and finally, Built in Exception, Custom exception, Throwable Class. **Programs:** 1. Create an Interface Stack with methods push(), pop() and show(). Create class IntStack which implements Stack. Demonstrate the working of this class. 2. Create an Interface Queue with methods insert(), delete() and display(). Create class StringQueue which implements Queue. Demonstrate the working of this class. 3. Analyze how a programmer must handle an error when an invalid mobile number is entered (lesser than 10 digits or greater than 10 digits; with alphabets; starting with 0) is entered while taking the details of a student. The Student class must contain the following: Name; USN; Branch and Mobile Number. Write a program using a custom exception type. 4. Create an exception called Non Int Result Exception, which is generated when the result of dividing two integer values produces a result with a fractional component. Non Int Result Exception has two fields that hold integer values; a constructor; and an override of the to String() method, allowing a friendlier description of the exception to be displayed using println(). 5. Create a Counter class with a private count instance variable and two methods. The first method: synchronized void increment() – tries to increment count by If count is already at its maximum of 3, then it waits until count is less than 3 before incrementing it. The other method is synchronized void decrement() – attempts to decrement count by 1. If count already at a minimum of 0, then it waits until count is greater than 0 before decrementing it. Every time either method has to wait, it displays a statement as to why it is waiting. Also every time an increment or decrement occurs, the Counter displays a statement that says what occurred and shows count's new value. 6. Create a Bank class with withdrawal and deposit methods. Demonstrate the use of notify() and wait().	**10**	**Apply**

Module – 4	No. of Hours	Blooms Cognitive Levels
Multithreaded Programming: Use of Multithread programming, Thread class and Runnable interface , Thread priority, Thread synchronization, Thread communication, Deadlock. **Swings:** Swings: The origins of Swing; Two key Swing features; Components and Containers; The Swing Packages; A simple Swing Application; Create a Swing J Applet; J label and Image Icon; J Text Field; The Swing Buttons; J Tabbed pane; J Scroll Pane; JList; J Combo Box; JTable. **Programs:** 1.Write a JAVA program which creates two threads, one thread displays "JAVA" 10 times, and another thread displays "PROGRAMMING LANGUAGE" 15 times continuously? 2.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. 3. Write a program that prints the Multiplication Table for numbers 2- 10. At a time the tables must print completely for one number before printing for the next number. 4. Create a GUI for a business of your choice using swings. Demonstrate the use of Button, Radio Button, Combo box, List, Textfield etc.	**10**	**Apply**
Module – 5		
Servlet : The Life Cycle of a Servlet; Using Tomcat for Servlet Development; A simple Servlet; The Servlet API; The Javax. servlet Package; Reading Servlet Parameter; The Javax. servlet.http package; Handling HTTP Requests and Responses; Using Cookies; Session Tracking. Java Server Pages (JSP): JSP, JSP Tags, Tomcat, Request String, User Sessions, Cookies, Session Objects. **Programs:** 1. Write Servlet application to print current date &time. 2. Write Servlet application to demonstrate session tracking. 3. Implement Program 1 using JSP. 4. Implement Program 2 using JSP.	**8**	**Apply**

Course outcomes: The students should be able to:

CO1: Demonstrate a solid understanding of object-oriented programming (OOP) principles and effectively apply them using Java.

CO2: Utilize core OOP constructs such as classes, inheritance, polymorphism, interfaces, and packages to develop modular and reusable code.

CO3: Implement robust exception handling and multithreading techniques to create efficient and error-resilient applications.

CO4: Design and develop Java-based GUI and web applications, incorporating servlets and event-driven programming to simulate real-world scenarios.

CO5: Apply Java syntax and semantics proficiently to write clean, maintainable, and scalable object-oriented programs.

Reference Books

1. Java: The Complete Reference, Herbert Schildt, 13th Edition, 2023, McGraw Hill Education, ISBN-13: 978-1265058431.
2. Java Programming: A Comprehensive Introduction, Herbert Schildt, Dale Skrien, 1st Edition, 2013, McGraw Hill Education, ISBN-13: 978-0078022074.
3. Programming with Java: A Primer, E. Balagurusamy, 5th Edition, 2014, McGraw Hill Education, ISBN-13: 978-9351343202.
4. Core Java Volume I – Fundamentals, Cay S. Horstmann, 13th Edition, 2024, Pearson Education, ISBN-13: 978-0138051808.
5. Head First Java: A Brain-Friendly Guide, Kathy Sierra, Bert Bates, Trisha Gee, 3rd Edition, 2021, O'Reilly Media, ISBN-13: 978-1491910771.

B.N.M. Institute of Technology

An Autonomous Institution under VTU

Department of Information Science and Engineering

SEMESTER–III

Course Name: Web Technology
Course Code: 24ISE136

L: T: P: J:	0:0:2:2	**CIA:**	50
Credit:	02	**SEA:**	50
Hours:	25	**SEA Duration:**	3 HOURS

Course Learning Objectives: The students will be able to

1. Understand the structure and semantics of HTML and create well-structured web pages.
2. Learn and apply core JavaScript concepts, including DOM manipulation and ES6 features, for interactive web applications.
3. Implement client-side scripting with JavaScript, event handling, and JSON for dynamic user interfaces.
4. Understand server-side programming using Java Servlets and JDBC for data handling and session management.
5. Analyze and apply React concepts and component-based architecture for building modern, scalable web applications.

PART A

Module-1	**No. of Hours**	**Blooms Cognitive Levels**
Introduction to HTML, what is HTML and Where did it come from? HTML Syntax, Semantic Markup, Structure of HTML Documents, HTML Elements, HTML5 Semantic Structure Elements. HTML Tables and Forms, Introducing Tables, Styling Tables, Introducing Forms, Form Control Elements, Table and Form Accessibility. **Build a Personal Portfolio Webpage, Create a Responsive Form, Style and Enhance an HTML Table.**	**5**	**Apply**
Module-2		
JavaScript: What is JavaScript and What can it do? JavaScript Design Principles, where does JavaScript Go? Syntax, JavaScript Objects, The Document Object Model (DOM), ES6 features, form validation and basic animation ES6 features, form validation and basic animation **Explore ES6 Features, Form Validation Exercise, Simple Animation with JavaScript.**	**5**	**Apply**
Module-3		
Client Side Programming Java Script: An introduction to JavaScript–JavaScript DOM Model-Exception HandlingValidation- Built-in objects-Event Handling- DHTML with JavaScript- JSON introduction –Syntax – Function Files. **Event-Driven Web App, Parse and Display JSON Data, DHTML Project.**	**5**	**Apply**
Module-4		
Server Side Programming Servlets: Java Servlet Architecture- Servlet Life Cycle- Form GET and POST actions- Session Handling- Understanding Cookies- DATABASE CONNECTIVITY: JDBC. **Self-Build a Login System Using Servlets, Form Handling with GET and POST, JDBC CRUD Operation**	**5**	**Apply**

Module-5	No. of Hours	Blooms Cognitive Levels
What is React? : Defining React, Why React? What problem does react Solve? React is not just another Framework, React Concept and Terminology. React Render HTML, React JSX, React Components, React Class, React PROPS **React Component Practice, JSX Practice Project, Props and State Demonstration**	**5**	**Apply**

PART –A (Programs) Not restricted to the list given

1. Design a web page that shows your class timetable using HTML table tag.
2. Design a form which should accept the student data such as, USN, Name of the Student, Date of birth, Branch, Semester. After submitting the form, It should display confirmation message.
3. Write a JavaScript to design a simple calculator to perform the following operations: Sum, product, difference, and quotient.
4. Write a JavaScript that calculates the squares and cubes of the numbers from 0 to 10 and outputs HTML text that displays the resulting values in an HTML table format.
5. Develop and demonstrate a HTML5 file that includes JavaScript script that uses functions for the following problems:
   a. To check whether a string is palindrome or not
   b. Fibonacci Sequence using Recursion.
6. Create a student registration form for job application and validate the form fields using JavaScript.
7. Develop a JavaScript program to shuffle deck of cards.
8. Write a program using react for creating a ToDo list.
9. Develop an Quiz App using React.
10. Create an currency app using React

PART - B (Mini - Project) Not restricted to the list given

Develop a web application project using the languages and concepts learnt in exercises listed in part A with a good look and feel effects. You can use any web technologies and frameworks and databases. Sample Projects Include : 1. Food Ordering Website 2. Online Purchase Store 3. Search Engine etc. 4. College website 5. Banking application 6. Blood donation application 7. Gaming application 8. Library management system 9. Chat application 10. Tourism website	**25**	**Apply**

Course Outcomes: After completing the course, the students will be able to	
CO 1	Understand the HTML semantics, JavaScript concepts, client-side scripting, server-side programming with servlets and JDBC, and the core principles of building modular front-end applications using React.
CO 2	Apply web development technologies including HTML5, JavaScript, DHTML, Servlets, JDBC, and React to design, validate the web applications with interactive and dynamic user interfaces.
CO 3	Design and develop the web applications by applying HTML semantics, implementing client-side functionalities using JavaScript and JSON, managing server-side operations with Java Servlets and JDBC, and building modern front-end interfaces using React.
CO 4	Analyse and implement server-side applications using Java Servlets and JDBC for session, cookie, and database interaction.

Reference Books:

1. Programming the World Wide Web, 8th Edition, Robert W. Sebesta, Pearson Education, ISBN: 978-0133775983, 2014.
2. Introduction to React, Cory Gackenheimer, 1st Edition, Apress, ISBN: 978-1484212462, September 2015.
3. Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics" by Jennifer Niederst Robbins, 5th Edition published in May 2018.
4. Learning React: Modern Patterns for Developing React Apps" by Alex Banks and Eve Porcello, Alex Banks and Eve Porcello, 2020
5. Learning React: A Hands-On Guide to Building Web Applications Using React and Redux, Kirupa Chinnathambi, 2nd Edition, Addison-Wesley Professional, ISBN: 978-0134843551, 2018.

Web links and Video Lectures:

1. https://academicearth.org/
2. https://onlinecourses.nptel.ac.in/noc20_ee98/preview
3. https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_01415769760627916821 1/overview

B.N.M. Institute of Technology

An Autonomous Institution under VTU, Approved by AICTE

Department of Mathematics

Syllabus

Semester: IV		
Course: Statistics, Probability and Graph theory **Course Code: 24MAC141 (Common to CSE, ISE, AIML)**		
L:T:P:J	**2:2:0:0**	**CIA: 50**
Credits:	**03**	**SEA: 50**
Hours:	**40**	**SEA Duration: 03 Hours**

Course Learning Objectives: The students will be able to

1	Provide an insight into applications of Graph Theory, Curve fitting & Statistical methods.
2	Develop the knowledge of probability, joint probability distribution and Queuing theory occurring in digital signal processing, design engineering and micro wave engineering.

Module-1: Curve fitting & Statistical methods	No. of hours	Blooms cognitive Levels
Examples from Engineering that require curve fitting and statistical methods. **Curve Fitting:** Curve fitting by the method of least squares-fitting the curves of the form: $y = ax+b$, y $= ax^b$ *and* $y = ax^2 + bx + c$. **Statistical methods:** Introduction to Moments, Skewness, Kurtosis and problems. Karl Pearson's coefficient of correlation and lines of regression. *Experiential Learning component: Problems on curve fitting and statistical methods*	**L: 04** **T: 04**	**L1** **L2** **L3**
Module-2: Probability distributions & Joint probability distribution		
Examples from Engineering that require Probability and Joint probability distribution **Probability distributions:** Review of basic probability theory. Discrete and continuous Random variables, probability mass/density functions (definitions only). Binomial, Poisson, exponential and normal distributions (without proof). **Joint probability distribution:** Joint Probability distribution for two discrete random variables, expectation, covariance and correlation. *Experiential Learning component: Problems on Binomial, Poisson, Exponential and Normal distributions*	**L: 04** **T: 04**	**L1** **L2** **L3**
Module-3:Markov chain & Sampling theory		
Examples from Engineering that require Markov Chain and Sampling Theory **Markov chain**: Introduction to Stochastic process, Probability vectors, Stochastic matrices, Regular stochastic matrices, Markov Chains, Higher transition probabilities, Stationary distribution of Regular Markov chains and absorbing states, Markovian processes. **Sampling theory**: Introduction to sampling theory, testing of hypothesis, level of significance, confidence limits, test of significance of mean and difference of means for large samples-z-test, test of significance of small Samples-Student's t- distribution. *Experiential Learning component: Problems on Markovian processes and, Sampling Theory*	**L: 04** **T: 04**	**L1** **L2** **L3**
Module-4: Queuing theory		
Examples from Engineering that require queueing theory Introduction, birth and death process, Kendall's Notation, Symbolic representation of a queuing model, single server Poisson queuing model with infinite capacity (M/M/1: ∞/FCFS), when $\lambda_n = \lambda$ and $\mu_n = \mu(\lambda < \mu)$, Multiple server Poisson queuing model with infinite capacity (M/M/S: ∞/ FCFS), when $\lambda_n = \lambda$ for all $n, (\lambda < S\mu)$, *Experiential Learning component: Problems on* (M/M/1: ∞/FCFS) and (M/M/S: ∞/ FCFS) queuing models	**L: 04** **T: 04**	**L1** **L2** **L3**
Module-5: Graph theory		
Examples from Engineering that require graph theory Basic concepts, types of graphs, order and size of a graph, in-degree and out-degree, bipartite-graphs, connected and disconnected graphs, Eulerian graph, Hamiltonian graphs, sub-graphs, isomorphic graphs. Matrix representation of graphs, adjacency matrix, incidence matrix. Planar graphs: definition, characterization of planar graphs, Kuratowski's theorem, Euler's formula and consequences. *Experiential Learning component: Problems on detection of planar and non-planar graphs*	**L: 04** **T: 04**	**L1** **L2** **L3**

Course Outcomes: After completing the course, the students will be able to

CO 1:	Make use of correlation and regression analysis to fit a suitable mathematical model for the statistical data.
CO 2:	Apply discrete and continuous probability and joint probability distributions in analyzing the probability models arising in engineering field.
CO 3:	Use Markov chain in prediction of future events and demonstrate the validity of testing the hypothesis.
CO 4:	Acquire skills in analyzing queuing models.
CO 5:	Apply the knowledge of Graph Theory in Network modeling, electrical network and computational algorithms.

CO - PO Mapping:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO 1	3	2			2							
CO 2	3	2			2							
CO 3	3	2			2							
CO 4	3	2			2							
CO 5	3	2			2							

Reference Books:

1. E. Kreyszig: "Advanced Engineering Mathematics", John Wiley & Sons, 10th Edition(Reprint), 2016.
2. B. S. Grewal: "Higher Engineering Mathematics", Khanna Publishers, 44th Edition, 2017.
3. S. D. Sharma : "Operations Research", Kedar Nath Ram Nath & Co. Meerut, 2014.
4. T. Veerarajan : Probability, Statistics and Random processes, McGraw Hill Education(India) Private Limited, Third edition, Nineteenth reprint 2017.
5. C. Ray Wylie, Louis C. Barrett : "Advanced Engineering Mathematics", 6th Edition, McGraw-Hill Book Co., New York, 1995.
6. B. V. Ramana: "Higher Engineering Mathematics" 11th Edition, Tata McGraw-Hill, 2010.
7. Srimanta Pal & Subodh C. Bhunia: "Engineering Mathematics", Oxford University Press, 3rd Reprint, 2016.
8. Narsingh Deo, "Graph Theory with Applications to Engineering and Computer Science", Prentice Hall of India, 2000.

Web links and Video Lectures:

1. https://nptel.ac.in/courses/111104098
2. https://www.youtube.com/watch?v=1YkfeR05YXY
3. https://archive.nptel.ac.in/courses/111/104/111104079/
4. https://www.youtube.com/watch?v=xGkpXk-AnWU
5. https://archive.nptel.ac.in/courses/106/104/106104170/

B.N.M. Institute of Technology

An Autonomous Institution under VTU

Department of Information Science and Engineering

SEMESTER–IV

Course: MICROCONTROLLER AND EMBEDDED SYSTEM
CourseCode:24ISE142

L:T:P:J	**3:1:1:0**	**CIA:**	**50**
Credits:	**04**	**SEA:**	**50**
Hours:	**50**	**SEA Duration:**	**3 HOURS**

Course Learning Objectives: The students will be able to

1. Understand the fundamentals of ARM based systems, basic hardware components, selectionmethods and attributes of an embedded system.
2. ProgramARM7controller using the various instructions.
3. Identify the applicability of the embedded system.
4. Comprehend the real time operating system used for the embedded system Develop and testProgramusingARM7TDMI/LPC2148.
5. Conduct the experiments on an ARM7TDMI/LPC2148 evaluation board using evaluation version ofEmbedded 'C' &Keil Uvision-4 tool/compiler.

Module-1	**No. of Hours**	**Blooms cognitiveLevels**
Microprocessors versus Microcontrollers, ARM7 Embedded Systems: The RISC design philosophy, The CISC design philosophy, The ARM Design Philosophy, Embedded System Hardware, Embedded System Software. ARM Processor Fundamentals: Registers, Current Program Status Register, Pipeline, Exceptions, Interrupts, and the Vector Table, Core Extensions.	10	Apply
Module-2		
Introduction to the ARM7 Instruction Set: Data Processing Instructions. Branch Instructions, Load-Store Instructions, Software Interrupt Instructions, Program Status Register Instructions, Loading Constants. Writing and Optimizing ARM Assembly Code: Writing Assembly code, Profiling and cycle counting, Instruction scheduling, Register Allocation, Conditional Execution, Looping Constructs. List of programs: 1. Write a program to add an array of 16 bit numbers & store the 32 bit result in internal RAM 2. Write a program to find the square of a number (1to10) using look-up table. 3. Write a program to find the largest number in an array of 32 numbers. 4. Write a program to arrange a series of 32 bit numbers in ascending order. 5. Write a program to count the number of ones and zeros in two consecutivememory locations.	10	Apply

Module-3		
Embedded System Components: Embedded Vs General computing system, History of embedded systems, Classification of Embedded systems, Major applications areas of embedded systems. **Typical Embedded System**: Core of an Embedded System, Memory, Sensors and Actuators (LED, 7segment LED display, stepper motor, Keyboard, Push button switch), Communication Interface (on board Communication Interface), Embedded firmware, Other system components. List of programs: 6. Display "Hello World" message using Internal UART. 7. Interface a Stepper motor and rotate it in clock wise and anti-clock wise direction.	10	Apply
Module-4		
Characteristics and Quality Attributes of an Embedded Systems: Characteristics of an Embedded Systems, Quality Attributes of an Embedded Systems. **Embedded Systems-Application and Domain Specific**: Washing Machine-Application Specific Embedded System. **Hardware and Software Co-Design and Program Modelling**: Fundamental Issues in Hardware Software Co-Design, Computational Models in Embedded Design, Introduction to Unified Modelling Language(UML). List of programs: 8. Demonstrate the use of an external interrupt to toggle an LED On/Off. 9. Display the Hex digits 0 to F on a 7-segment LED interface, with anappropriate delay in between.	10	Apply
Module-5		
RTOS and IDE for Embedded System Design: How to choose an RTOS, Integration and testing of Embedded hard ware and firmware, Embedded system Development Environment–Block diagram (excluding Keil), Disassembler / decompiler, simulator, emulator and debugging techniques, target hardware debugging, boundary scan. List of programs: 10. Demonstrate the use of an external interrupt to toggle an LED On/Off. Display the Hex digits 0 to F on a 7-segment LED interface, with an appropriate delay in between.	10	Apply

Course Outcomes: After completing the course, the students will be able to	
CO 1	Describe the architectural features and instructions of ARM7 microcontroller
CO 2	Apply the knowledge gained for Programming ARM7 for different applications and InterfaceExternal devices and I/O with ARM7 microcontroller.
CO 3	Interpret the basic hardware components and their selection method based on theCharacteristics and attributes of an embedded system.
CO 4	Develop the hardware/software co-design and firmware design approaches.
CO 5	Demonstrate the need of real time operating system for embedded system applications
CO 6	Develop and test program using ARM7 TDMI/LPC2148

Reference Books:

1. ARM System Developer's Guide: Designing and Optimizing System Software, Andrew N. Sloss, Dominic Symes, Chris Wright, 1st Edition, Elsevier/Morgan Kaufmann, 2004, ISBN-13: 978-1558608740.
2. Introduction to Embedded Systems, K.V. Shibu, 2nd Edition, McGraw Hill Education, 2016, ISBN-13: 978-9339219680.
3. Microcontroller (ARM) and Embedded Systems, Raghunandan G.H., 1st Edition, Cengage Learning India Pvt. Ltd., 2020, ISBN-13: 978-9353504106.
4. The Insider's Guide to the Philips ARM7-Based Microcontrollers: An Engineer's Introduction to the LPC2100 Series, Trevor Martin, 1st Edition, Hitex, 2005, ISBN-13: 978-0954998813.
5. ARM System-on-Chip Architecture, Steve Furber, 2nd Edition, Pearson Education, 2000, ISBN-13: 978-0201675191.
6. Embedded Systems: Architecture, Programming and Design, Raj Kamal, 2nd Edition, Tata McGraw-Hill Education, 2008, ISBN-13: 978-0070151253.

Web links and Video Lectures:

1. https://academicearth.org/
2. https://onlinecourses.nptel.ac.in/noc20_ee98/preview

B.N.M. Institute of Technology

An Autonomous Institution under VTU

Department of Information Science & Engineering

Semester: IV

Course Name: Database Management System		**Course Code: 24ISE143**
L: T: P: J	3:0 :1:1	**CIA Marks:** 50
Credits:	4	**SEA Marks:** 50
Hours/Week (Total)	5	**SEA Duration:** 03 Hours

Course Learning Objectives: The students will be able to

1	Understand fundamental concepts, terminology and application of databases, SQL and NoSQL
2	Design concepts and creation of relational databases using relation algebra.
3	Practice SQL programming through a variety of database problems.
4	Demonstrate the use of Normalization, concurrency and transactions in database.

Module-1: Database System Concepts, Data Modeling	**No. of Hours**	**Blooms cognitive Levels**
Databases and Databases Users: Characteristics of database Approach, Advantages of using the DBMS Approach. Database System Concepts and Architecture: Data Models-Schemas and Instances, Three-Schema Architecture and Data Independence, Database Languages and Interfaces. Data Modeling Using the Entity-Relationship (ER) Model: Entity Types-Entity sets- Attributes and Keys, Relationship types – Relationship Sets – Roles and structural Constraints, Weak Entity Types. **Practical component:** Draw ER Diagram for the following Databases using GitMind software. Order Database Library Database Bank Database	10	**Understand CO1**
Module-2: Relational Data Model and Relational Algebra		
Concept of relations, schema-instance distinction, keys, referential integrity and foreign keys, relational algebra operators: selection, projection, cross product, various types of joins, division, example queries,converting the database specification in E/R notation to the relational schema **Practical component:** Create Schema, insert at least 5 records in each table and add appropriate constraints for the following Library Database using ORACLE or MySQL DBMS under LINUX/Windows environment BOOK (Book_id, Title, Publisher_Name, Pub_Year) BOOK_AUTHORS (Book_id, Author_Name) PUBLISHER (Name, Address, Phone) BOOK_COPIES (Book_id, Branch_id, No-of_Copies) BOOK_LENDING (Book_id, Br_id, Card_No, Date_Out, Due_Date) LIBRARY_BRANCH (Branch_id, Branch_Name, Address) Write SQL queries to 1. Retrieve details of all books in the library – id, title, name of publisher,authors, number of copies in each branch, etc. 2. Get the particulars of borrowers who have borrowed more than 3 books,but from Jan 2020 to Jun 2022. 3. Delete a book in BOOK table. Update the contents of other tables toreflect this data manipulation operation.	**10**	**Apply CO2**

Module-3: SQL		
Basic SQL: SQL Data Definition and Data Types, Specifying Constraints in SQL, Basic Retrieval Queries in SQL, INSERT – DELETE and UPDATE Statements in SQL, Additional features in SQL **More SQL: Complex Queries, Triggers, Views and Schema Modification:** Complex SQL Retrieval Queries, Specifying Constraints as Assertions and actions as Triggers, Views (Virtual Tables) in SQL. **Practical component:** Create Schema, insert at least 5 records for each table and add appropriate constraints for the following Order Database using ORACLE or MySQL DBMS under LINUX/Windows environment. SALESMAN (Salesman_id, Name, City, Commission) CUSTOMER (C_id, Cust_Name, City, Grade, Salesman_id) ORDERS (Ord_No, Purchase_Amt, Ord_Date, C_id, S_id) Write SQL queries to 1. Count the customers with grades above Bangalore's average. 2. Find the name and numbers of all salesman who had more than onecustomer. 3. List all the salesman and indicate those who have and don't have customers in their cities (Use UNION operation.) 4. Create a view that finds the salesman who has the customer with the highest order.	**10**	**Apply CO3**
Module-4: Functional Dependencies and Normalization		
Basics of Functional Dependencies and Normalization for Relational Database: Functional Dependencies, Armstrong's axioms for FD's, Equivalent Decompositions, closure of a set of FDs, minimal covers, Normal forms Based on Primary Keys, General Definitions of Second and Third Normal Forms, Boyce- Codd Normal Forms [BCNF] **Practical component:** Create Schema, insert at least 5 records for each table and add appropriate constraints for the following Company Database using ORACLE or MySQL DBMS under LINUX/Windows environment. EMPLOYEE (SSN, Name, Address, Sex, Salary, Super SSN, D No) DEPARTMENT (D No, D Name, Mgr. SSN, Mgr. Start Date) DLOCATION(D No,D Loc) PROJECT (P No, P Name, P Location, D No) WORKS_ON(SSN, P No, Hours) Write SQL queries to 1. Make a list of all project numbers for projects that involve an employee whose last name is 'Scott', either as a worker or as a manager of the department that controls the project. 2. Show the resulting salaries if every employee working on the 'IoT' project is given a 10 percent raise. **3.** Find the sum of the salaries of all employees of the 'Accounts' department, as well as the maximum salary and the average salary in this department.	**10**	**Analyze CO4**

Module-5: Transaction Processing, Concurrency Control, NoSQL		
Introduction to Transaction Processing –Introduction to Transaction Processing, Desirable Properties on Transactions (ACID) **Concurrency Control Techniques:** Transactions and Schedules, Serializability, Precedence Graphs, Concurrency, Lock Based Protocols: 2PL, Strict 2PL Protocols, Deadlocks - Detection and Prevention **NoSQL:** SQL v/s NoSQL, The Emergence of NoSQL, BASE Properties, Data Models: Relationships, Graph Database, Schema less Database.	**10**	**Analyze CO5**

Course Outcomes: After completing the course, the students will be able to	
24ISE143.1	Understand the Database System Concepts along with Data Modeling Using the Entity-Relationship (ER) Model
24ISE143.2	Apply the concepts of relations on RDBMS, constraints, joints using relational algebra operators.
24ISE143.3	Apply Structured Query Language for database manipulation.
24ISE143.4	Analyze functional dependencies to normalize relations of relational database
24ISE143.5	Analyze transactions processing, schedules protocols, serializability issues, deadlocks in DBMS and concepts of NoSQL with its advantages

Reference Books

1. **Fundamentals of Database Systems**, Ramez Elmasri, Shamkant B. Navathe, 7th Edition, 2015, Pearson Education, ISBN-13: 978-0133970777.
2. **Database System Concepts**, Abraham Silberschatz, Henry F. Korth, S. Sudarshan, 7th Edition, 2019, McGraw-Hill Education, ISBN-13: 978-0078022159.
3. **NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence**, Pramod J. Sadalage, Martin Fowler, 1st Edition, 2012, Addison-Wesley Professional, ISBN-13: 978-0321826626.

B.N.M. Institute of Technology

An Autonomous Institution under VTU

Department of Information Science and Engineering

Design and Analysis of Algorithms

SEMESTER– IV

Subject Code: 24ISE144	L:T:P:J: 3:0:2:0	CIE Marks:	50
Credits:	4	SEE Marks:	50
Total Number of Lecture Hours	50	SEE Duration:	3 Hours

Course objectives:

- Analyze the asymptotic performance of algorithms.
- To understand the concept of designing an algorithm.
- Synthesize efficient algorithms in common engineering design situations.

Analyze the efficiency of programs based on time complexity.

Module 1	Teaching Hours	Blooms Cognitive Levels
Introduction: Notion of algorithm, Fundamentals of Algorithmic Problem Solving, Fundamentals of the Analysis of Algorithmic Efficiency: Analysis frame work, Asymptotic Notations and BasicEfficiency Classes, Mathematical Analysis of Non-recursive and Recursive Algorithms. **Practical Programs:** 1. Implement the following concepts: a. Linear Search b. To find the maximum and minimum from a given list of n elements using Brute Force Method. 2. There are 5 books in the shelf, find the number of ways to select 3 books from 5 books on the shelf using the NCR with recursion. 3. Find the next three terms of the sequence 15, 23, 38, 61, … Fibonacci series of the given number using recursion.	**10**	**Analyze**
Module 2		
Brute Force: Selection Sort and Bubble Sort, Sequential Search and Brute Force String Matching **Divide and Conquer**: General method, Binary search, Recurrence equation for divide and conquer, Finding the Maximum and Minimum, Merge sort, Quick sort, Strassen's matrix multiplication. Decrease and Conquer Approach: Topological Sort. **Practical Programs:** 1. Implement the Selection sort algorithm. 2. Implement Bubble sort algorithm. 3. Implement the Sequential Search algorithm. 4. Write a program to search a key in a given set of elements using Binary search method and find the time required to find the key. 5. Write a program to find Maximum and Minimum using divide and conquer technique and find the time required to find the elements. 6. Sort a given set of elements using Merge Sort method and determine the time required sort the elements. Plot a graph of number of elements versus time taken. Specify the time efficiency class of this algorithm. 7. Sort a given set of elements using Quick Sort method and determine the time	**10**	**Apply**

required sort the elements. Plot a graph of number of elements versus time taken. Specify the time efficiency class of this algorithm. 8. Implement Topological sort using source removal method find the time required to sort the elements.		
Module 3		
Greedy Method: General method, Coin Change Problem, Knapsack Problem, Job sequencing with deadlines. Minimum cost spanning trees: Prim's Algorithm, Kruskal's Algorithm. Single source shortestpaths: Dijkstra's Algorithm. Optimal Tree problem: Huffman Trees and Codes. **Practical Programs:** 1. Implement Coin Changing problem method and find the time required. 2. Implement Job Sequence problem using Greedy method. 3. Implement Prim's algorithm and Find Minimum Cost Spanning Tree of a given connected undirected graph. 4. Implement Kruskal's algorithm and Find Minimum Cost Spanning Tree of a given connected undirected graph. 5. Implement Dijkstra's algorithm find shortest paths to other vertices from a given vertex in a weighted connected graph.	**10**	**Apply**
Module 4		
Dynamic Programming: General method with Examples, Multistage Graphs. Transitive Closure: Warshall's Algorithm, All Pairs Shortest Paths: Floyd's Algorithm, Optimal Binary Search Trees, Knapsack problem, Bellman-Ford Algorithm, Travelling Sales Person problem, Reliability design. **Practical Programs:** 1. Implement all-pairs shortest paths problem using Floyd's algorithm. 2. Implement all-pairs shortest paths problem using Warshal's algorithm. 3. Implement 0/1 Knapsack using Dynamic Programming. 4. Implementation of Bellman Ford Algorithm using a directed graph. 5. Implement Travelling Sales man problem using Dynamic Programming.	**10**	**Apply**
Module 5		
Backtracking: General method (T2:7.1), N-Queens problem, Sum of subsets problem, Graph coloring, Hamiltonian cycles. Branch and Bound: Assignment Problem, Travelling Sales Person problem, 0/1 Knapsack problem: LC Branch and Bound solution, FIFO Branch and Bound solution. NP- Complete and NP-Hard problems: Basic concepts, nondeterministic algorithms, P, NP, NP-Complete and NP-Hard classes **Practical Programs:** 1. Implementation of N Queen Problem using Backtracking technique. 2. Implementation of SUM-SUBSET Problem. 3. Design and implement to find all Hamiltonian Cycles in a connected undirected Graph G of n vertices using the backtracking principle. 4. Implementation Assignment Problem using Backtracking technique. 5. Implementation of Travelling Sales Man Problem using Branch and Bound method. 6. Implementation of Knapsack problem using Branch and Bound method.	**10**	**Analyze**

Course outcomes: The students should be able to:	
CO1	Apply and Analyze the asymptotic runtime complexity of algorithms by using mathematical relations that helps to identify them in specific instances
CO2	Apply and solve problems using brute force, divide and conquer techniques
CO3	Apply various problem-solving methodologies such as greedy, decrease and conquer to solve a given problem.
CO4	Apply the dynamic programming to estimate the computational complexity of different algorithms.
CO5	Apply and Analyze the efficient algorithm design approaches in a problem specific manner in terms of space and time complexity

Reference Books:

1. Introduction to Algorithms, Thomas H. Cormen, Charles E. Leiserson, Ronal L. Rivest, Clifford Stein, 4th Edition, MIT Press, April 2022.
2. Introduction to the design and analysis of algorithms, by Anany Levitin, 3rd Edition, Pearson Education, 2021.
3. Fundamentals of Computer Algorithms, Ellis Horowitz, Sartaj Sahni, Sanguthevar Rajasekaran, 2nd Edition, Universities Press, 2008
4. The Art of Computer Programming, Donald E. Knuth, Addison-Wesley, Volume 4B (2023)
5. Algorithms Unlocked, Thomas H. Cormen, MIT Press, 2013

Web Links:

https://www.cs.duke.edu/courses/fall08/cps230/Book.pdf.

https://www.tutorialspoint.com/design_and_analysis_of_algorithms/design_and_analysis_of_algorithms_tutorial.pdf.

B.N.M. Institute of Technology

An Autonomous Institution under VTU

Department of Information Science & Engineering

Semester: IV

Course: Robotic Process Automation

Course Code: 24ISE145

L:T:P:J	**0:0:2:2**	**CIA**	**:**	**50**
Credits:	**02**	**SEA**	**:**	**50**
Hours:	**24**	**SEA Duration**	**:**	**03 Hours**

Course Learning Objectives: The students will be able to

1	To Understand the basic concepts and platforms of RPA.
2	To Describe the different types of variables and its Datatypes.
3	To Describe the various types of Sequence and Control flow.
4	To Apply various control techniques.
5	To Apply Screen Scraping and OCR in RPA along with Error Handling.

Descriptions (if any):

Prerequisite

- **CPU with 1.4GHz 32-bit is minimum requirement** and **recommended is Dual Core 1.8GHz 64-bit.**
- **4GB** is the **minimum requirement and 8GB** is the **recommended** RAM.
- **Windows 7 or above** is required **OS**, though recommended is **Windows 10.**

Part A

Module-1:	No. of hours	Blooms Cognitive Levels
RPA Foundations and Platforms: What is RPA, Components of RPA, RPA Platforms- About UiPath - The future of automation, Record and PIay, Downloading and installing UiPath Studio, Learning UiPath Studio, Task recorder **DOWNLOADING AND INSTALLING UIPATH STUDIO.** On UiPath Installation, download the Community Cloud version. **REVERSING A STRING** **A Quick Guide To The Top RPA Tool:** Using Project Templates and UiPath Studio Components **"WELCOMING TO THE SESSION "** **Introduction to UiPath and its Components:** Using Types of Projects and Templates in UiPath and UiPath Components	**L: 04**	**Apply**
Module-2:		
Variables and Data Types-Variables and Scope, Collections, Data Types, Arguments, Purpose and use, File operation with step-by-step example-CSV/Excel **A) Display a Message in Message Box directly** **B) Display Message in a Message Box using Variables** C) **Assign Activity**: To assign a value to a variable.	**L: 05**	**Apply**

Example-Count Number of Files D) **Write CSV Activity**: To save the specified Data Table to a .csv file.		
Module-3:		
Sequence and Control Flow-Sequencing the workflow, Activities-Control flow, various types of loops and decision making-Step-by Step example using Sequence. **ACTIVITIES IN UIPATH** **If Activity:** Find whether a number is even/odd **For Each Activity:** Print Fibonacci Series **While Activity:** Print Numbers 1-10 **Do While Activity:** Print Numbers 1-10 **Switch Activity**: Sum of two numbers is even/odd	**L: 05**	**Apply**
Module-4:		
Taking Control of the Controls : Finding and attaching windows , Act on controls - mouse and keyboard activities **ACTIONS** **A) WINDOWS ACTIONS** To automate the action of getting the title of an active window. **B) MOUSE CLICKS** To automate the action of closing a notepad window. **C) KEYSTROKES** To automate the task of writing text into a notepad file.	**L: 05**	**Apply**
Module-5:		
Screen Scraping and OCR- Screen Scrapping, When to use OCR, Types of OCR available, How to use OCR, **WEB EXTRACTION-WEB SCRAPING OF GOOGLE CONTACTS-** Extract data from Google Contacts and store it in a file. **READ PDF WITH OCR ACTIVITY** Show the uses of optical character recognition to scan the images inside the PDF document and output all the text as a Variable **EMAIL ACTIVITIES** Read the emails and filter the emails with sender name, subject, and body of the email.	**L: 05**	**Apply**

PART – B (Mini projects)

The List of Possible Projects

1. **MOVING FILES FROM SOURCE FOLDER TO DESTINATION FOLDER**
Task: The aim is to automate the process of moving files from the source folder to the destination folder.

2. WEB AUTOMATION
Task: The aim is to scrape data from a website and store it in a .csv File.

3. EMAIL AUTOMATION
Task: The aim is to save the attachments of the top 30 emails which have a keyword in the subject line. The attachments will be stored in a particular folder.

4. EXCEL AUTOMATION
Task: The aim is to fill a google form from the data stored in a .csv file automatically.

5. PDF AUTOMATION
Task: The aim is to extract text and images from a PDF File and store the output in a Message Box/TextFile.

6. IMAGE & TEXT AUTOMATION
Task: Consider an application is running in a virtual environment. Now, once we enter the account number, amount, choose transaction type, a transaction ID is automatically generated.Our task is to, display the Transaction ID in a message box once the transaction is successfully completed

7. AUTOMATED CUSTOMER SUPPORT EMAILS (WITH INSTRUCTIONS)
Task: The aim is to send automated replies to emails that have particular text in their subject line. For example, If a subject line has the word 'refund,' your solution would send an automated reply to it accordingly.

8. TO FIND OUT WHETHER A NUMBER IS EVEN OR ODD.
Task: The aim is to find out whether a number is even or odd using the different components of UiPath Studio and Dashboard Of UiPath

9. EXTRACTING DATA FROM AN E-COMMERCE WEBSITE
Task: Is to design automation which reads the subject line of an email and performs a custom search. So,
For example, if my subject line says LG TV then, the designed automation must search for LG TVs' in Flipkart and extract the name, URL, and the price.

10. HOW TO AUTOMATE TASKS IN A VIRTUAL ENVIRONMENT?
Task: Automating Task on a Desktop Application:
Consider a scenario where you want to open a Notepad in the virtual environment and want to type something in it. How will you automate this task?

****Projects are not limited**

Course Outcomes:
After studying these laboratory programs using UIPath, students will be able to

1. To Understand the basic concepts and platforms of RPA.
2. To Describe the different types of variables and its Datatypes.
3. To Describe the various types of Sequence and Control flow.
4. To Apply various control techniques.
5. To Apply Screen Scraping and OCR in RPA along with Error Handling.

Text Books:

1. **The Robotic Process Automation Handbook: A Guide to Implementing RPA Systems**, Tom Taulli, 1st Edition, February 2020, Apress, ISBN-13: 978-1484257296.
2. **Learning Robotic Process Automation: Create Software Robots and Automate Business Processes with the Leading RPA Tool – UiPath**, Alok Mani Tripathi, 1st Edition, March 2018, Packt Publishing, ISBN-13: 978-1788470940.

Reference Books:

1. **Introduction to Robotic Process Automation: A Primer**, Frank Casale, Rebecca Dilla, Heidi Jaynes, Lauren Livingston, 1st Edition, 2015, Institute of Robotic Process Automation, ISBN: Not available.
2. **Robotic Process Automation: Guide to Building Software Robots, Automate Repetitive Tasks & Become an RPA Consultant**, Richard Murdoch, 1st Edition, 2018, Independently Published, ISBN-13: 978-1983036835.
3. **Robotic Process Automation Tools, Process Automation and Their Benefits: Understanding RPA and Intelligent Automation**, Srikanth Merianda, 1st Edition, 2018, CreateSpace Independent Publishing Platform, ISBN-13: 978-1720626077.

Web links

- https://www.uipath.com/rpa/robotic-process-automation
- https://www.uipath.com
- https://academy.uipath.in
- https://www.edureka.co/blog