BNM Institute of Technology

An Autonomous Institution under VTU, Approved by AICTE

Department of Mathematics

Syllabus

Semester: III			
Course: Fourier Transform, Fundamentals of logic and Linear Algebra **Course Code: 22MAI131 (Common to CSE, ISE, AIML)**			
L:T:P:J	**2:1:1:0**	**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 arising in engineering

Module-1: Fourier Series & Fourier Transforms	**No. of hours**	**Blooms cognitive Levels**
Examples from Engineering field 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. ***Lab Component:*** *Finding the Fourier series and Fourier Transform of a function*	**L : 04** **T : 04**	**CO1** **Apply**
Module-2: Fundamentals of logic and Relations		
Examples from Engineering field that require Fundamentals of logic and Relations. **Fundamentals of logic:** Basic connectives and truth tables, logic equivalence - the laws of logic, logical implication- rules of inference **Relations:** First order linear recurrence relation, second order linear homogenous recurrence relation with constant coefficients. ***Lab Component:*** *Finding the solution of recurrence relation*	**L : 04** **T : 04**	**CO2** **Apply**
Module-3: Vector Spaces		
Examples from Engineering field 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. ***Lab Component:*** *problems on linearly independent and dependent, basis and dimension of a vector space.*	**L : 04** **T : 04**	**CO3** **Apply**
Module-4: Linear Transformation		
Examples from Engineering field 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. ***Lab Component:*** *problems on Inverse of a linear transformation and Rank-nullity theorem*	**L : 04** **T : 04**	**CO4** **Apply**
Module-5: Inner Product Spaces		
Examples from Engineering field 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. ***Lab Component:*** *Problems on QR-factorization and singular value decomposition*	**L : 04** **T : 04**	**CO5** **Apply**

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: Communicate the basic concepts of logic and their relevance for computer science engineering.

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.

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. H. K. Dass, " Advanced Engineering Mathematics" S. Chand publication.
4. C.Ray Wylie, Louis C.Barrett : "Advanced Engineering Mathematics", 6" Edition, 2. McGraw-Hill Book Co., New York, 1995.
5. James Stewart : "Calculus —Early Transcendentals", Cengage Learning India Private Ltd., 2017.
6. B.V.Ramana: "Higher Engineering Mathematics" 11th Edition, Tata McGraw-Hill, 2010.
7. Srimanta Pal & Subobh C Bhunia: "Engineering Mathematics", Oxford University Press, 3"Reprint, 2016.
8. David C. Lay, Steven R. Lay and J. J. McDonald "Linear Algebra and its applications", 3rd Edition, Pearson Education Ltd., 2017.
9. 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=cHNmT1-qurk
6. https://www.youtube.com/watch?v=ATqV_I8DCh0

Assessment Process:

<table>
<tr><th rowspan="2">PCC</th><th rowspan="2">CIA</th><th rowspan="2">SEA</th><th colspan="3">CIA (50)</th><th rowspan="2">SEA
Conduction: 100 M
Reduced to: 50 M</th></tr>
<tr><th></th><th>I</th><th>II</th></tr>
<tr><td rowspan="4">Conduction</td><td rowspan="4">50</td><td rowspan="4">50</td><td rowspan="2">Written Test</td><td>50</td><td>50</td><td rowspan="4">Five questions with each of 20 marks (with internal choice). Student should answer one full question from each module</td></tr>
<tr><td colspan="2">Average of two tests – scaled down to 25 Marks</td></tr>
<tr><td>Assignment</td><td colspan="2">Two assignments – one for 10 marks and another for 5 marks = 15</td></tr>
<tr><td>AAT</td><td colspan="2">10</td></tr>
<tr><td colspan="4">Total – 50 marks</td><td>Total – 50 marks</td></tr>
</table>

Note: Minimum 40% passing marks in all divisions

BNM Institute Of Technology

Dept. of Computer Science & Engineering

Choice Based Credit System (CBCS and Outcome Based Education (OBE)

Semester:III

Course Name: Computer Organization		**Course Code: 22CSE132**
L: T: P: J	3:0:0:0	**CIA Marks:** 50
Credits:	3	**SEA Marks:** 50
Hours/Week (Total)	3 (40)	**SEA Duration:** 03 Hours

Course Learning Objectives: The students will be able to

1	To understand the basic sub systems of a computer, their organization, structure, and operation.
2	Illustrate the concept of programs as sequences of machine instructions.
3	Teach the concepts of Memory system and cache memory.
4	Cultivate clear thinking in performing Arithmetic, Multiplication, division, and Floating-point number operations in computer.
5	Describe the working of pipelining and multiprocessor computer architecture.

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 CO1**
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 CO2**
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**	**Analyze CO3**
Module 4: MIPS Arithmetic operations		
MIPS Addressing for 32-Bit Immediates and Addresses, Parallelism, and Instructions: Synchronization, Translating and Starting a Program, Addition and Subtraction (MIPS), Multiplication and Division (MIPS).	**8**	**Apply CO4**
Module 5:Pipelining and Multiprocessors		
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**	**Apply CO5**

Course Outcomes: After completing the course, the students will be able to	
22CSE132.1	Ability to understand the abstraction of various components of a computer.
22CSE132.2	Ability to understand the functions of different sub systems, such as processor, Input/output, and memory
22CSE132.3	Analyze the concepts of Memory system and cache memory.
22CSE132.4	Apply Arithmetic, Multiplication, and division operations in computer.
22CSE132.5	Apply the working of pipelining and multiprocessor computer architecture.

Text Books

1. Carl Hamacher, ZvonkoVranesic, SafwatZaky, Computer Organization, 5th Edition, Tata McGraw Hill, 2002.
2. Parallel Programming for Multicore and Cluster Systems, Thomas Rauber, Gudula Runger, 2nd Edition, Springer, 2013.
3. David A. Patterson and John L. Hennessey, "Computer organization and design, The Hardware/Software interface", Morgan Kauffman / Elsevier, Fourth/Fifth edition, 2014.

Reference Books

1. William Stallings: Computer Organization & Architecture, 9th Edition, Pearson, 2015.

Marks Distribution for Assessment:

<table>
<tr><th rowspan="2">PCC</th><th rowspan="2">CIA</th><th rowspan="2">SEA</th><th colspan="3">CIA (50)</th><th rowspan="2">SEA
Conduction: 100 M
Reduced to: 50 M</th></tr>
<tr><th></th><th>I</th><th>II</th></tr>
<tr><td rowspan="4">Conduction</td><td rowspan="4">50</td><td rowspan="4">50</td><td rowspan="2">Written Test</td><td>50</td><td>50</td><td rowspan="4">Five questions with each of 20 marks (with internal choice). Student should answer one full question from each module</td></tr>
<tr><td colspan="2">Average of two tests – scaled down to 25 Marks</td></tr>
<tr><td>Assignment</td><td colspan="2">Two assignments – one for 10 marks and another for 5 marks = 15</td></tr>
<tr><td>AAT</td><td colspan="2">10</td></tr>
<tr><td colspan="4">Total – 50 marks</td><td>Total – 50 marks</td></tr>
</table>

Note: Minimum 40% passing marks in all divisions

BNM Institute Of Technology

Dept. of Computer Science & Engineering

Choice Based Credit System (CBCS and Outcome Based Education (OBE)

Semester: III

Course Name: Operating Systems		**Course Code: 22CSE133**
L:T:P:J	**2: 1:1:0**	**CIAMarks:**50
Credits:	**3**	**SEAMarks:**50
Hours/Week (Total)	**4 (40)**	**SEADuration:**03Hours

Course Learning Objectives: The students will be able to

1	Introduce concepts and terminology used in OS
2	Explain threading and multithreaded systems
3	Illustrate process synchronization and concept of Deadlock
4	Introduce to Unix File Systems

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.	**8**	**Apply CO1**
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.	**8**	**Apply CO2**
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, Garbage Collection, Bad Block Management.	**8**	**Apply CO3**
Module-4: UNIX file system		
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 pathnames. File related commands – cat, mv, rm, cp, wc and od commands. **Practical component**: Execution of UNIX Shell Commands	**8**	**Apply CO4**
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,	**8**	**Analyze CO5**

egrep. **Shell programming**: Ordinary and environment variables. Read and read-only commands. Command line arguments. exit and exit status of a command. 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		

Course Outcomes: After completing the course, the students will be able to	
22CSE133.1	Apply the concepts of process scheduling to improve CPU utilization and identify various multi- threading models
22CSE133.2	Identify the need of policies, protection required in managing deadlock, main and virtual memory & various techniques in managing concurrent processes
22CSE133.3	Apply the concept of paging & segmentation for effective memory management
22CSE133.4	Apply the concepts of Unix system and file commands to perform various tasks in files and system.
22CSE133.5	Analyze the concepts of Wildcards and Shell Programming to write basic shell scripts and formulating regular expressions for Pattern matching

Text Books

1. Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, Operating System Principles 7th edition, Wiley-India, 2006
2. Sumitabha Das., Unix Concepts and Applications., 4thEdition., Tata McGraw Hill .

Reference Books

1. William Stallings, Operating System: Internals and Design Principles, Prentice Hall, 9th Edition, 2018.
2. W. Richard Stevens: Advanced Programming in the UNIX Environment, 2nd Edition, Pearson Education, 2005
3. Unix System Programming Using C++ - Terrence Chan, PHI, 1999.

Marks Distribution for Assessment:

<table>
<tr><th rowspan="2">PCC</th><th rowspan="2">CIA</th><th rowspan="2">SEA</th><th colspan="3">CIA (50)</th><th rowspan="2">SEA
Conduction: 100 M
Reduced to: 50 M</th></tr>
<tr><th></th><th>I</th><th>II</th></tr>
<tr><td rowspan="4">Conduction</td><td rowspan="4">50</td><td rowspan="4">50</td><td rowspan="2">Written Test</td><td>50</td><td>50</td><td rowspan="4">Five questions with each of 20 marks (with internal choice). Student should answer one full question from each module</td></tr>
<tr><td colspan="2">Average of two tests – scaled down to 25 Marks</td></tr>
<tr><td>Assignment</td><td colspan="2">Two assignments – one for 10 marks and another for 5 marks = 15</td></tr>
<tr><td>AAT</td><td colspan="2">10</td></tr>
<tr><td colspan="4">Total – 50 marks</td><td>Total – 50 marks</td></tr>
</table>

Note: Minimum 40% passing marks in all divisions

BNM Institute of Technology

Dept. of Computer Science and Engineering

Choice Based Credit System (CBCS and Outcome Based Education (OBE)

Semester: III

Course Name: Data Structure & Applications **Course Code: 22CSE134**

L: T: P: J	3:0:2:0	**CIA Marks:** 50
Credits:	4	**SEA Marks:** 50
Hours/Week (Total)	5 (50)	**SEA Duration:** 03 Hours

Course Learning Objectives: The students will be able to

1	Learn the fundamental data structures and identify data structuring strategies that are appropriate for a given contextual problem.
2	Design, develop, test and debug in C language considering appropriate data structure.
3	Illustrate and implement basic data structures such as stack, queue and linked list and apply them for the given problem.
4	Understand and distinguish the conceptual and applicative differences in trees, binary trees and binary search trees. Apply the concepts of trees for the given application.
5	Create and use appropriate data structures in C programs for solving real life problems.

Module-1: Introduction to DS, Stacks and Queues	**No. of Hours**	**Blooms Cognitive Levels with CO mapping**
Introduction to DS: 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. **Sample Programs:** 1. Write a C program to implement data structure. 2. Write C Program to convert the given infix expression to postfix expression. 3. Write a C Program to Evaluate the given postfix Expression. 4. Write a C Program to implement Queue data structure. 5. Write a C Program to implement circular Queue data structure.	**10**	**Apply CO1, CO2, CO3**
Module-2: Linked List – I		
Linked Lists: Definition, Create, Insert, Delete, Update, Traverse, and Position-based Operations, Concatenate, Merge, and Reverse Lists, Doubly Linked List Implementation and Operations. **Sample Programs:** 1. Write a C Program to perform following operations on Singly Linked List 1. a. Create b. Insert (At the Mid Position) c. Delete (At End) d. Display 2. Write a C Program to store and display the specified student information using SLL. 3. Design, Develop and Implement a menu driven Program in C for the following operations on Doubly Linked List (DLL) of Employee Data with	**10**	**Apply CO1, CO2, CO3**

the fields: SSN, Name, Dept, Designation, Sal, PhNo a. Create a DLL of N Employees Data by using end insertion. b. Display the status of DLL and count the number of nodes in it c. Perform Insertion and Deletion at End of DLL d. Perform Insertion and Deletion at Front of DLL e. Demonstrate how this DLL can be used as Double Ended Queue. f. Exit		
Module-3: Linked List – II & Trees		
Circular Linked List Implementation and Operations, Applications of Lists (Polynomial addition). Implementation of stacks and queues using Linked List **Introduction to Trees:** General Tree Representation, Traversals, Applications. **Sample Programs:** 1. Write a program to implement stack using SLL. 2. Write a program to implement Queue using SLL. 3. Develop a C code to perform polynomial addition using circular Singly Linked List with header node. 4. Develop a code to traverse a tree.	**10**	**Apply CO1, CO2, CO3**
Module-4: Advanced Trees and Heap		
Binary Trees: Definition, Properties, Traversals, Applications. **Binary Search Tree**: Definition, Implementation, Search, Insert, Delete operations. Building and Evaluating Binary Expression Tree. **Heap:** Definition, Implementation, Insert, Delete, Peek operations. **Sample Programs:** 1. Design, Develop and Implement a menu driven Program in C for the following operations on Binary Search Tree (BST) of Integers. a. Create a BST of N Integers: 6, 9, 5, 2, 8, 15, 24, 14, 7, 5, 2. b. Traverse the BST in Inorder, Preorder and Postorder. c. Search the BST for a given element (KEY) and report the appropriate message. d. Exit. 2. Design, Develop and Implement a code to generate a max and min heap tree.	**10**	**Apply CO1, CO2, CO3, CO4**
Module-5: Hashing & Graphs		
Hashing: Hash Table, Hash Functions, Collision Handling by Open Addressing, Chaining. **Graphs:** Disjoint sets, Representation of Graphs - Adjacency/ Cost Matrix, Adjacency Lists, and Traversal of Graphs (BFS and DFS). **Sample Programs:** 1. Given a File of N employee records with a set K of Keys (4-digit) which uniquely determine the records in file F. Assume that file F is maintained in memory by a Hash Table (HT) of m memory locations with L as the set of memory addresses (2-digit) of locations in HT. Let the keys in K and addresses in L are Integers. Design and develop a Program in C that uses Hash function H: K →L as H(K)=K mod m (remainder method) and implement hashing technique to map a given key K to the address space L.	**10**	**Apply CO1, CO2, CO3, CO4, CO5**

Resolve the collision (if any) using linear probing. 2. Design, Develop and Implement a Program in C for the following operations on Graph(G) of Cities a. Create a Graph of N cities using Adjacency Matrix. b. Print all the nodes reachable from a given starting node in a digraph using BFS method. 3. Print all the nodes reachable from a given starting node in a digraph using DFS method.		

Course Outcomes: After completing the course, the students will be able to	
22CSE134.1	Apply fundamental data structures viz., Lists, Stacks, Queues, Linked Lists, Binary Trees from first principles.
22CSE134.2	Identify the use of appropriate data structures for a given problem.
22CSE134.3	Design and implement solutions to basic practical problems using customized data structures.
22CSE134.4	Apply the Advanced concepts like Heap & Hashing to solve problems.
22CSE134.5	Apply the concepts to solve graphical problems.

Text Books

1. "Data Structures and Program Design in C", Robert Kruse, C L Tondo, Bruce Leung and Shashi Mogalla, PHI, 2nd Edition, 2015.
2. Y. Langasm, M. J. Augenstein, A. M. Tenenbaum (2001) Data Structures Using C and C++, Prentice Hall India, New Delhi, India.

Reference Books

1. T. H. Cormen, C. E. Leiserson and R. L. Rivest (1990) Introduction to Algorithms, Third Edition, MIT Press, MA.
2. Fundamentals of Data Structures in C -- by Horowitz, Sahni and Anderson-Freed (Silicon Press 2007).
3. Data Structures and Algorithm Analysis in C++, Mark Allen Weiss, 4th Revised edition; 2013, Addison-Wesley, ISBN-13: 978-8131714744
4. Data Structures Using C, Reema Thareja, 1st Edition, 2011, Oxford Higher Education, ISBN-13: 978-0198099307.

Marks Distribution for Assessment:

<table>
<tr><th rowspan="2">PCI</th><th rowspan="2">CIA</th><th rowspan="2">SEA</th><th colspan="3">CIA (50)</th><th colspan="2">SEA
Conduction: 100 M
Reduced to: 50 M</th></tr>
<tr><th></th><th>I</th><th>II</th><th>PART A</th><th>PART B</th></tr>
<tr><td rowspan="3">Conduction</td><td rowspan="3">50</td><td rowspan="3">50</td><td rowspan="2">IA Test</td><td>30</td><td>30</td><td rowspan="2">30 Marks</td><td rowspan="2">70 Marks</td></tr>
<tr><td colspan="2">Average of two tests – 30 M</td></tr>
<tr><td>Continuous Assessment</td><td colspan="2">weekly Assessment -20 marks</td></tr>
<tr><td colspan="5">Total – 50 Marks</td><td colspan="2">Total – 50 Marks</td></tr>
</table>

i) CIA: 50%

IA Test: 2 IA tests - each of 30 Marks	Average of 2 tests – 30 M
Practical Lab record – 10 Marks Performance – 05 Marks Viva – 05 Marks	20 Marks
	Total 50 Marks

ii) SEA : 50%

Question Paper:

Theory part	5 questions to answer each of 6 Marks 2 questions from each module with internal choice Student should answer one full question from each module	6 M x 5 = 30 Marks
Execution part	Write up - 20 Marks Conduction - 40 Marks Viva-Voce - 10 Marks	70 Marks
	Total	**100 Marks reduced to 50 M**

Note:

- **No Assignment and AAT**
- **Minimum 40% passing marks in all divisions**

B.N.M. Institute of Technology

Dept. of Computer Science & Engineering

Choice Based Credit System (CBCS and Outcome Based Education (OBE)

Logic Design and Applications

SEMESTER–III

Course Name: Logic Design and Applications **Course Code: 22CSE135**

L: T: P: J	3:0:2:0	CIE Marks: 50
Credits:	4	SEE Marks: 50
Hours/Week (Total)	5 (50)	SEE Duration: 3 Hours

Course Objectives:

- Explain the use of Operational Amplifier, 555 timer IC, Regulator ICs and uA741, Analog-to-Digital and Digital-to-Analog conversion techniques.
- Make use of simplifying techniques in the design of combinational circuits.
- Illustrate combinational and sequential digital circuits
- Demonstrate the use of flip flops and apply for registers

Module-1: Analog Electronics	**Teaching Hours**	**Blooms cognitive Levels with CO mapping**
Analog Electronics: Introduction to Operational Amplifier, Operational Amplifier Application Circuits: Multivibrators using IC-555, Peak Detector, Schmitt trigger, Non-Linear Amplifier, Relaxation Oscillator, Current-to-Voltage and Voltage-to-Current Converter, Regulated Power Supply Parameters, adjustable voltage regulator , D to A and A to D converter. ***Laboratory Component:*** 1. Design and simulate a 1 kHz Relaxation Oscillator with 50% duty cycle using ua 741 Opamp 2. Design and simulate an astable multivibrator circuit for three cases of duty cycle (50%, <50% and >50%) using NE 555 timer IC. 3. Design and simulate a Schmitt trigger for given UTP and LTP using ua 741 opamap.	**10**	**CO1 Understand**
Module-2: Digital Electronics		
Digital Electronics: Review of Basic Logic gates, Positive and Negative Logic, Introduction to HDL. **Combinational Logic Circuits:** Sum-of-Products Method, Truth Table to Karnaugh Map, Pairs Quads, and Octets, Karnaugh Simplifications, Don't-care Conditions, Product-of-sums Method, Product-of-sums simplifications, Simplification by Quine-McClusky Method, Hazards and Hazard covers, HDL Implementation Models. ***Laboratory Component:*** Given a 4-variable logic expression, simplify it using appropriate technique and implement the same using basic gates	**10**	**CO2 Apply**
Module-3: Data-Processing Circuits		
Data-Processing Circuits: Multiplexers, Demultiplexers, 1-of-16 Decoder, BCD to Decimal Decoders, Seven Segment Decoders, Encoders, Exclusive-OR Gates, Parity Generators and Checkers, Magnitude Comparator, Programmable Array Logic, Programmable Logic Arrays, HDL Implementation of Data Processing Circuits. Arithmetic Building Blocks, Arithmetic Logic Unit. ***Laboratory Component:*** 1. Given a 4-variable logic expression, simplify it using appropriate technique and realize the simplified logic expression using 8:1 multiplexer IC. 2. Design and implement: i) Parity Generator (ii) Parity Checker	**10**	**CO3 Analyze**

Module-4: Flip- Flops		
Flip- Flops: RS Flip-Flops, Gated Flip-Flops, Edge-triggered RS FLIP-FLOP, Edge-triggered D FLIP-FLOPs, Edge-triggered JK FLIP-FLOPs. FLIP-FLOP Timing, JK Master-slave FLIP-FLOP, Switch Contact Bounce Circuits, Various Representation of FLIP-FLOPs, HDL Implementation of FLIP-FLOP. **Registers:** Types of Registers, Serial In - Serial Out, Serial In - Parallel out, Parallel In - Serial Out, Parallel In - Parallel Out, Universal Shift Register, Applications of Shift Registers, Register implementation in HDL. ***Laboratory Component:*** Realize a J-K Master / Slave Flip-Flop using NAND gates and verify its truth table. And implement the same in HDL.	**10**	**CO4 Analyze**
Module-5: Counters		
Counters: Asynchronous Counters, Decoding Gates, Synchronous Counters, Changing the Counter Modulus. Decade Counters, Presettable Counters, Counter Design as a Synthesis problem, A Digital Clock, Counter Design using HDL. ***Laboratory Component:*** 1. Design and implement a mod-n (n<8) synchronous up counter using J-K Flip-Flop ICs and demonstrate its working. 2. Design and implement an asynchronous counter using decade counter IC to count up from 0 to n (n<=9) and demonstrate on 7-segment display (using IC-7447)	**10**	**CO5 Analyze**

Course Outcomes: After completing the course, the students will be able to	
22CSE135.1	Interpret with the applications of analog circuits using Operational Amplifier
22CSE135.2	Apply Karnaugh Map, and Quine-McClusky Methods to simplify digital circuits.
22CSE135.3	Analyze the combinational logic circuits and simulate using HDL
22CSE135.4	Analyze the sequential logic circuits with different types of flip-flops and simulate using HDL.
22CSE135.5	Design and Analyze the functionalities of registers and counters

Text Books

1. Donald P Leach, Albert Paul Malvino & Goutam Saha, Digital Principles and Applications, 8th Edition, Tata McGraw Hill, 2015.
2. Charles H Roth Jr, Larry L Kinney and Raghunandan G. H. Analog and Digital Electronics, Cengage Learning, 2020.

Reference Books

1. M. Morris Mano, Digital Design, 4th Edition, Pearson Prentice Hall, 2008.
2. David A. Bell, Electronic Devices and Circuits, 5th Edition, Oxford University Press, 2008

Marks distribution for assessment

<table>
<tr><th rowspan="2">PCI</th><th rowspan="2">CIA</th><th rowspan="2">SEA</th><th colspan="3">CIA (50)</th><th rowspan="2">SEA
Conduction: 100 M
Reduced to: 50 M</th></tr>
<tr><th></th><th>I</th><th>II</th></tr>
<tr><td rowspan="5">Conduction</td><td rowspan="5">50</td><td rowspan="5">50</td><td rowspan="2">Written Test</td><td>50</td><td>50</td><td rowspan="5">Five questions with each of 20 marks (with internal choice). Student should answer one full question from each module</td></tr>
<tr><td colspan="2">Average of two tests – 50 marks scaled down to 15 marks</td></tr>
<tr><td>Assignment</td><td colspan="2">Average of 2 Assignments – 10M</td></tr>
<tr><td>Practical</td><td colspan="2">Weekly Assessment – 10 Marks
IA test – 15 Marks
(IA test to be conducted for 50 M and scaled down to 15M)</td></tr>
<tr><td colspan="5">Total – 50 Marks</td><td>Total – 50 Marks</td></tr>
</table>

i) CIA: 50%

Theory	**IA Test** (Theory): 2 IA tests - each of 50 Marks – Average of 2 tests scaled down to 15 Marks **Assignment** : 2 Assignments – each of 10 marks	25 Marks
Lab	**Weekly Assessment** – 10 Marks **Practical test (1)** - 15 marks	25 Marks
	Total	**50 arks**

ii) SEA : 50%

Question Paper:

Theory Exam	5 questions to answer, each of 20 Marks 2 questions from each module with internal choice Student should answer one full question from each module	20 M x 5 = **100 M** **Reduced to 50 M**
	Total	**50 Marks**

Note: Minimum 40% passing marks in all divisions

B N M Institute of Technology

Dept. of Computer Science &Engineering

Choice Based Credit System (CBCS and Outcome Based Education (OBE)

Semester: III

Course Name: Object Oriented Programming Using JAVA **Course Code: 22CSE136**

L:T:P:J	0:0:2:2	CIA Marks: 50
Credits:	2	SEA Marks: 50
Hours/Week (Total)	4 (30)	SEA Duration: 3 Hours

Pre-Requisites:

Course Learning Objectives : The students will be able to

1	Learn fundamental features of object oriented language and JAVA
2	Set up Java JDK environment to create, debug and run simple Java programs.
3	Create multi-threaded programs.
4	Solve real world problems using JAVA.

Module-1	No. of Hours	Blooms cognitive Levels
Introduction to Java: Features of OOP, Characteristics/Buzz words of Java, Java Environment: JDK, JVM, JRE, Fundamental Programming Structure in Java, Variables, Data Types, Operators & Expressions, Control Statements, Iteration Statements, Command Line Arguments, Arrays.	**6**	**Understand CO1**
Module-2		
Classes & Objects: Defining Classes & Objects, Access Specifies, Constructors, Overloading Constructor, Method Overloading, Passing and Returning object form Method, new operator, finalize() method, this keyword, Static Keyword, Encapsulation, Polymorphism. **Inheritance:** Defining a Inheritance, Types of Inheritance, Constructor in subclass, Method Overriding, super keyword, abstract keyword, final keyword.	**6**	**Understand CO2**
Module - 3		
Interfaces & Packages: Defining a Interface, Implementing a Interface, Difference between Interface & Classes, Extending a Interface, Usage of Package, Classpath, Importing a Package. **Exceptions:** Definition of Exception, Classification of Exception, Structure of Try & catch block, Error Vs Exception, Throw Keyword, Throws Keyword, Finally Keyword, Custom Exception	**6**	**Apply CO3**
Module-4		
Multi Threaded Programming: Multi Threaded Programming: What are threads? How to make the classes threadable ; Extending threads; Implementing runnable; Synchronization; Changing state of the thread; Bounded buffer problems, producer consumer problems.	**6**	**Apply CO4**
Module-5		
Database Connectivity: Basic SQL, Introduction to JDBC, JDBC Architecture & Drivers, Create a Database, Table, CRUD Operations, Java Application Using JDBC Connectivity, Driver Manager, ResultSet, Connection, Statement, Prepared Statement, DB Connectivity Steps, Store & Retrieving Image in SQL, JDBC CRUD Application.	**6**	**Create CO5**

Course Outcomes: After completing the course, the students will be able to	
22CSE136.1	Understand object oriented programming concepts and implement in java.
22CSE136.2	Comprehend building blocks of OOPs language, inheritance, package and interfaces.
22CSE136.3	Identify exception handling methods.
22CSE136.4	Implement multithreading in OOP.
22CSE136.5	Create solutions for real world problems using JAVA

Text Books

1. The Complete Reference, Java 2 (Fourth Edition), Herbert Schild, - TMH.
2. Java Fundamentals A comprehensive introduction By Herbert Schildt, Dale Skrien, McGraw Hill Education.
3. Programming with Java A Primer – E.Balaguruswamy, McGrawhill
4. Core Java Volume-I Fundamentals Horstmann& Cornell, - Pearson Education. - Eight Edition
5. Head First Java: A Brain-Friendly Guide, 2nd Edition- Kathy Sierra, Bert Bates

Marks Distribution for Assessment:

<table>
<tr><th>PBL</th><th>CIA</th><th>SEA</th><th colspan="3">CIA (50)</th><th>SEA
Conduction: 100 M
Reduced to: 50 M</th></tr>
<tr><td rowspan="4">Conduction</td><td rowspan="4">50</td><td rowspan="4">50</td><td rowspan="3">Theory</td><td>I IA</td><td>II IA</td><td rowspan="3">Project
Assessed for 100 marks
reduced to 50 Marks</td></tr>
<tr><td>25</td><td>25</td></tr>
<tr><td colspan="2">Average of 2 tests – 25 M</td></tr>
<tr><td>Practical</td><td colspan="2">Weekly Assessment
(Record/Project) – 10 Marks
Lab IA test – 15 Marks</td><td></td></tr>
<tr><td colspan="4">Total – 50 Marks</td><td>Total – 50 Marks</td></tr>
</table>

SEA : 50%

Project	Write up – 10 Marks Project report – 25 Marks Presentation & Demonstration - 50 Marks Viva-Voce – 15 Marks	100 Marks Reduced to 50 Marks
	Total	**50 Marks**

Note: Minimum 40% passing marks in all divisions

B.N.M. Institute of Technology

An Autonomous Institution under VTU

Semester: III		
COURSE: Soft Skill-1		
Course Code: 22SFT138	**L:T:P:J: 0:0:2:0**	**CIA Marks:** 100
Credits:	**1**	
Hours:	**24 hrs**	

Course Learning Objectives: The students will be able	
1	To help students understand their strengths and weakness.
2	To develop analytical and creative ability to solve problems individually or as a team.
3	To make students industry ready through practice of corporate etiquettes.
4	To enhance public speaking and presentation skills.

Module No.	Contents of the Module	Hours	Cos
1	**Module-1 Understanding and Managing Self** Self-Awareness, Self-Management, Anger Management, Time management, Change management. Vision and goal setting - Diff between vision and goal, smart, stretched goal concept, case studies Knowledge, Skill, Attitude Personality analysis using Big 5 personality test Critical Thinking, Problem solving, Creativity and innovation Integrity, ethics, values	8	**1 &2**
2	**Module -2 Corporate etiquettes and Mannerism** Introduction to Etiquette and Mannerism, Personal Etiquette, Grooming etiquettes- professional styling, Body & personality styling, Video Interview Etiquettes, Personal Interview EtiquettesEffective meeting skills. Workplace behavior, Personal interview	6	**3**
3	**Module -3 Public Speaking and presentation skills** Introduction to public speaking, making ideas, illustrating and delivering ideas, overcoming fear of public speaking and developing great delivery. Advanced Business presentation skills, PowerPoint presentation, Group discussion	**6**	**4**
4	**Module -4 Team Work** Interpersonal skills, group work vs team work	**4**	**5**

Course Outcomes: At the end of the course the student will be able to:

CO1	Understand their strength and weaknesses.
CO2	Develop analytical and creative ability to solve problems.
CO3	Identify themselves as industry ready through the practice of corporate etiquettes.
CO4	Enhance public speaking and presentation skills.
CO5	Build team collaboration by working towards shared goals.

Mapping of Course Outcomes with Programme Outcomes:

COs	PO8	PO9	PO10	PO11
CO1	3	3		
CO2		3		3
CO3	3	3		3
CO4		3	3	
CO5			3	3

MOOC Course:

Communicate with impact - https://www.coursera.org/learn/communicate-with-impact

Leading Diverse Teams - https://www.coursera.org/learn/leading-diverse-teams

Practical component:

1. Mock GD and interview may be conducted at the end of the course to check their confidence. Students can prepare their SWOT analysis and present the same.
2. The students are to be involved in various activities and games such as Just a Minute or Pick and speak to demonstrate each topic.

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: 22MAI141 (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**	**Apply**
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**	**Apply**
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, Goodness of fit-Chi-Square test. *Experiential Learning component: Problems on Markovian processes and, Sampling Theory*	**L: 04** **T: 04**	**Apply**
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**	**Apply**
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**	**Apply**

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.

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. James Stewart : Calculus-Early Transcendental, Cengage Learning India Private Ltd., 2017.
7. B. V. Ramana: "Higher Engineering Mathematics" 11th Edition, Tata McGraw-Hill, 2010.
8. Srimanta Pal & Subodh C. Bhunia: "Engineering Mathematics", Oxford University Press, 3rd Reprint, 2016.
9. 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/

Assessment Process:

<table>
<tr><th rowspan="2">PCC</th><th rowspan="2">CIA</th><th rowspan="2">SEA</th><th colspan="3">CIA (50)</th><th rowspan="2">SEA
Conduction: 100 M
Reduced to: 50 M</th></tr>
<tr><th></th><th>I</th><th>II</th></tr>
<tr><td rowspan="4">Conduction</td><td rowspan="4">50</td><td rowspan="4">50</td><td rowspan="2">Written Test</td><td>50</td><td>50</td><td rowspan="4">Five questions with each of 20 marks (with internal choice). Student should answer one full question from each module</td></tr>
<tr><td colspan="2">Average of two tests – scaled down to 25 Marks</td></tr>
<tr><td>Assignment</td><td colspan="2">Two assignments – one for 10 marks and another for 5 marks = 15</td></tr>
<tr><td>AAT</td><td colspan="2">10</td></tr>
<tr><td colspan="4">Total – 50 marks</td><td>Total – 50 marks</td></tr>
</table>

Note: Minimum 40% passing marks in all divisions

BNM Institute of Technology

Dept. of Computer Science and Engineering

Choice Based Credit System (CBCS and Outcome Based Education (OBE)

Semester: IV

Course Name: Microcontroller and Embedded Systems

Course Code: 22CSE142

L: T: P: J	2:1:1:0	**CIA Marks:** 50
Credits:	**3**	**SEA Marks:** 50
Hours/Week (Total)	**4 (40)**	**SEA Duration:** 03 Hours

Pre-Requisites: -

Course Learning Objectives: The students will be able to

1	Understand the fundamentals of ARM-based systems, including programming modules with registers and the CPSR.
2	Use the various instructions to program the ARM controller.
3	Program various embedded components using the embedded C program.
4	Identify various components, their purpose, and their application to the embedded system's applicability.

Module1: Microprocessors versus Microcontrollers	**No. of Hours**	**Blooms Cognitive Levels with CO mapping**
Microprocessors versus Microcontrollers, ARM Embedded Systems: The RISC 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.	**6**	**Understand CO1**
Laboratory Component: Using Keil software, observe the various registers, dump, CPSR, with a simple ALP programme.	**2**	
Module2: ARM Instruction Set		
Introduction to the ARM Instruction Set: Data Processing Instructions, Branch Instructions, Software Interrupt Instructions, Program Status Register Instructions, Coprocessor Instructions, Loading Constants.	**6**	**Apply CO2**
Laboratory Component: 1. Write a program to find the sum of the first 10 integer numbers. 2. Write a program to find the factorial of a number.	**2**	
Module-3: ARM programming using Assembly language		
ARM programming using Assembly language: Writing Assembly code, Profiling and cycle counting, instruction scheduling, Register Allocation, Conditional Execution, Looping Constructs.	**6**	**Apply CO3**
Laboratory Component: 1. Write a program to add an array of 16-bit numbers and store the 32-bit result in internal RAM. 2. Write a program to find the square of a number (1 to 10) using a look-up table.	**2**	

3. Write a program to find the largest or smallest number in an array of 32 numbers. 4. Write a program to arrange a series of 32 bit numbers in ascending/descending order. 5. Write a program to count the number of ones and zeros in two consecutive memory locations		
Module4: I/O Subsystems and Hardware Interface		
Sensors, Actuators, LED, 7 segment LED display, DAC, ADC, Stepper motor, Keyboard, Push button switch, Communication Interface. ***Laboratory Component:*** 1. Interface and Control a DC Motor. 2. Interface a Stepper motor and rotate it in clockwise and anti-clockwise direction. 3. Interface a DAC and generate Triangular and Square waveforms. 4. Interface ADC to obtain Digital output for a given Analog input using Internal ADC of ARM controller. 5. Display the Hex digits 0 to F on a 7-segment LED interface, with an appropriate delay in between.	**4** **4**	**Apply CO4**
Module-5: Embedded System Components and Embedded Environment		
Embedded System Components: Embedded Vs General computing system, History of embedded systems, Classification of Embedded systems, Major applications areas of embedded systems, purpose of embedded systems. Core of an Embedded System including all types of processor/controller, Memory **Embedded system Development Environment** – Block diagram (excluding Keil), Disassembler/decompiler, simulator, emulator and debugging techniques, target hardware debugging, boundary scan.	**8**	**Apply CO5**

Course Outcomes: After completing the course, the students will be able to	
22CSE142.1	Understand the fundamentals of ARM-based systems, including programming modules with registers and the CPSR.
22CSE142.2	Make use of Instruction sets and addressing modes learnt to write simple programs.
22CSE142.3	Apply the knowledge gained for Programming ARM controller for real time applications.
22CSE142.4	Apply the knowledge to interface external devices and I/O with ARM microcontroller.
22CSE142.5	Identify the importance of Embedded Systems in real time.

Text Books
1. Andrew N Sloss, Dominic Symes and Chris Wright, ARM system developers guide, Elsevier, Morgan Kaufman publishers, 2008. 2. Shibu K V, "Introduction to Embedded Systems", Tata McGraw Hill Education, Private Limited, 2nd Edition.
Reference Books

1. Raghunandan. G.H, Microcontroller (ARM) and Embedded System, Cengage learning Publication,2019
2. The Insider's Guide to the ARM7 Based Microcontrollers, Hitex Ltd.,1st edition, 2005.
3. Steve Furber, ARM System-on-Chip Architecture, Second Edition, Pearson, 2015.
4. Raj Kamal, Embedded System, Tata McGraw-Hill Publishers, 2nd Edition, 2008.

Marks Distribution for Assessment:

<table>
<tr><th rowspan="2">PCI</th><th rowspan="2">CIA</th><th rowspan="2">SEA</th><th colspan="3">CIA (50)</th><th rowspan="2">SEA
Conduction: 100 M
Reduced to: 50 M</th></tr>
<tr><th></th><th>I</th><th>II</th></tr>
<tr><td rowspan="5">Conduction</td><td rowspan="5">50</td><td rowspan="5">50</td><td rowspan="2">Written Test</td><td>50</td><td>50</td><td rowspan="4">Five questions with each of 20 marks (with internal choice). Student should answer one full question from each module</td></tr>
<tr><td colspan="2">Average of two tests – 50 marks scaled down to 15 marks</td></tr>
<tr><td>Assignment</td><td colspan="2">Average of 2 Assignments – 10M</td></tr>
<tr><td>Practical</td><td colspan="2">Weekly Assessment – 10 Marks
IA test – 15 Marks
(IA test to be conducted for 50 M and scaled down to 15M)</td></tr>
<tr><td colspan="4">Total – 50 Marks</td><td>Total – 50 Marks</td></tr>
</table>

i) CIA: 50%

Theory	**IA Test** (Theory): 2 IA tests - each of 50 Marks – Average of 2 tests scaled down to 15 Marks **Assignment** : 2 Assignments – each of 10 marks	5 Marks
Lab	**Weekly Assessment** – 10 Marks **Practical test (1)** - 15 marks	5 Marks
	Total	**50 arks**

ii) SEA : 50%

Question Paper:

Theory Exam	5 questions to answer, each of 20 Marks questions from each module with internal choice Student should answer one full question from each module	20 M x 5 = **100 M reduced to 50 M**
	Total	**50 Marks**

Note: Minimum 40% passing marks in all divisions

BNM Institute of Technology

Dept. of Computer Science & Engineering

Choice Based Credit System (CBCS) and Outcome Based Education (OBE)

Semester: IV

Course Name: Database Management System		**Course Code: 22CSE143**
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, tuple relation calculus, domain relational calculus, 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.	**10**	**Apply CO2**

3. Delete a book in BOOK table. Update the contents of other tables toreflect this data manipulation operation.		
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 one customer. 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	
22CSE143.1	Understand the Database System Concepts along with Data Modeling Using the Entity-Relationship (ER) Model
22CSE143.2	Apply the concepts of relations on RDBMS, constraints, joints using relational algebra operators.
22CSE143.3	Apply Structured Query Language for database manipulation.
22CSE143.4	Analyze functional dependencies to normalize relations of relational database
22CSE143.5	Analyze transactions processing, schedules protocols, serializability issues, deadlocks in DBMS and concepts of NoSQL with its advantages

Text Books	
1.	Ramez Elmasari, Shamkant B Navathe "Fundamentals of Database Systems", Pearson, Seventh Edition 2017.
2.	"Database System Concepts", Silberschatz, H Korth, S Sudarshan, 6th Edition, McGraw-Hill, 2010
3.	Pramod J Sadalage, Martin Fowler, "NOSQL Distilled", Pearson, 2013

Marks Distribution for Assessment:

PCI	CIA	SEA	CIA (50)			SEA Conduction: 100 M Reduced to: 50 M
				I	II	
Conduction	50	50	Written Test	50	50	Five questions with each of 20 marks (with internal choice). Student should answer one full question from each module
				Average of two tests – 50 marks scaled down to 15 marks		
			Assignment	Average of 2 Assignments – 10M		
			Practical	eekly Assessment – 10 Marks IA test – 15 Marks (IA test to be conducted for 50 M and scaled down to 15M)		
			Total – 50 Marks			**Total – 50 Marks**

i) CIA: 50%

Theory	**IA Test** (Theory): 2 IA tests - each of 50 Marks – Average of 2 tests scaled down to 15 Marks **Assignment** : 2 Assignments – each of 10 marks	25 Marks
Lab	**Weekly Assessment** – 10 Marks **Practical test (1)** - 15 marks	25 Marks
	Total	**50 arks**

ii) SEA : 50%

Question Paper:

Theory Exam	5 questions to answer, each of 20 Marks questions from each module with internal choice Student should answer one full question from each module	20 M x 5 = **100 M** **reduced to 50 M**
	Total	**50 Marks**

Note: Minimum 40% passing marks in all divisions

BNM Institute of Technology

Dept. of Computer Science & Engineering

Choice Based Credit System (CBCS and Outcome Based Education (OBE)

Semester: IV

Course Name: Design and Analysis of Algorithms		**Course Code: 22CSE144**
L: T: P: J	**3 : 0 :2 :0**	**CIA Marks:** 50
Credits:	**4**	**SEA Marks:** 50
Hours/Week (Total)	**5 (50)**	**SEA Duration:** 03 Hours

Pre-Requisites:

Course Learning Objectives: The students will be able to

1	Analyze the asymptotic performance of algorithms.
2	To understand the concept of designing an algorithm.
3	Synthesize efficient algorithms in common engineering design situations.
4	Analyze the efficiency of programs based on time complexity

Module-1:	**No. of 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 Basic Efficiency Classes, Mathematical Analysis of Non-recursive and Recursive Algorithms. **Practical Programs:** 1. Implement Coin Changing problem method and find the time required. 2. Write a program to find maximum profit using Knapsack technique. 3. Implement Job Sequence problem using Greedy method. Apply brute force/divide and conquer technique to recursively implement the following concepts: a. Linear Search or Binary Search. b. To find the maximum and minimum from a given list of n elements using Brute Force Method. 4. 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. 5. Find the next three terms of the sequence 15, 23, 38, 61, … Fibonacci series of the given number using recursion. 6. Demonstrate through a program how a sequence of characters is taken and checked for the possibility of the presence of the required string. If the possibility is found then, character matching is performed else no using Rabin Karp method.	**6 hours (Theory) 4 hours (Practical)**	**Apply CO1**
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.	**6 hours (Theory) 4 hours (Practical)**	**Apply CO2**

Practical Programs: 1. Implement the Selection sort algorithm. 2. Implement Bubble sort algorithm. 3. Implement the Sequential Search algorithm. 4. Implement the String Matching algorithm. 5. 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. 6. Write a program to find Maximum and Minimum using divide and conquer technique and find the time required to find the elements. 7. 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. 8. Sort a given set of elements using Quick 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. 9. 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 shortest paths: Dijkstra's Algorithm. Optimal Tree problem: Huffman Trees and Codes. **Practical Programs:** 1. Implement Coin Changing problem method and find the time required. 2. Write a program to find maximum profit using Knapsack technique. 3. Implement Job Sequence problem using Greedy method. 4. Implement Prim's algorithm and Find Minimum Cost Spanning Tree of a given connected undirected graph. 5. Implement Kruskal's algorithm and Find Minimum Cost Spanning Tree of a given connected undirected graph. 6. Implement Dijkstra's algorithm find shortest paths to other vertices from a given vertex in a weighted connected graph.	**6 hours (Theory) 4 hours (Practical)**	**Apply CO3**
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.	**6 hours (Theory) 4 hours (Practical)**	**Apply CO4**

5. Implement Travelling Sales man problem using Dynamic Programming.		
Module-5:		
General method (T2:7.1), N-Queens problem, Sum of subsets problem, Graph coloring, Hamiltonian cycles. Programme and Bound: Assignment Problem, Travelling Sales Person problem, 0/1 Knapsack problem: LC Programme and Bound solution, FIFO Programme 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.	**6 hours (Theory) 4 hours (Practical)**	**Apply CO5**

Course Outcomes: After completing the course, the students will be able to	
22CSE144.1	Apply and Analyze the asymptotic runtime complexity of algorithms by using mathematical relations that helps to identify them in specific instances.
22CSE144.2	Apply and solve problems using brute force, divide and conquer techniques
22CSE144.3	Apply various problem solving methodologies such as greedy, decrease and conquer to solve a given problem.
22CSE144.4	Apply the dynamic programming to estimate the computational complexity of different algorithms.
22CSE144.5	Apply and Analyze the efficient algorithm design approaches in a problem specific manner in terms of space and time complexity

Text Books
1. Introduction to the Design and Analysis of Algorithms, Anany Levitin: 2rd Edition, 2009. Pearson. 2. Computer Algorithms/C++, Ellis Horowitz, Satraj Sahni and Rajasekaran, 2nd Edition, 2014, Universities Press.
Reference Books
1. Introduction to Algorithms, Thomas H. Cormen, Charles E. Leiserson, Ronal L. Rivest, CliffordStein, 3rd Edition, PHI. 2. Data Structures and Algorithms using C, R.S.Salaria, 5th Edition, Khanna Publication.

Marks Distribution for Assessment:

<table>
<tr><th rowspan="2">PCI</th><th rowspan="2">CIA</th><th rowspan="2">SEA</th><th colspan="3">CIA (50)</th><th colspan="2">SEA
Conduction: 100 M
Reduced to: 50 M</th></tr>
<tr><th></th><th>I</th><th>II</th><th>PART A</th><th>PART B</th></tr>
<tr><td rowspan="4">Conduction</td><td rowspan="4">50</td><td rowspan="4">50</td><td rowspan="2">IA Test</td><td>30</td><td>30</td><td rowspan="3">30 Marks</td><td rowspan="3">70 Marks</td></tr>
<tr><td colspan="2">Average of two tests – 30 M</td></tr>
<tr><td>Continuous Assessment</td><td colspan="2">Weekly Assessment -20 marks</td></tr>
<tr><td colspan="3">Total – 50 Marks</td><td colspan="2">Total – 50 Marks</td></tr>
</table>

i) CIA: 50%

IA Test: 2 IA tests - each of 30 Marks	Average of 2 tests – 30 M
Practical Lab record – 10 Marks Performance – 05 Marks Viva – 05 Marks	20 Marks
	Total 50 Marks

ii) SEA : 50%

Question Paper:

Theory part	5 questions to answer each of 6 Marks questions from each module with internal choice Student should answer one full question from each module	6 M x 5 = 30 Marks
Execution part	Write up - 20 Marks Conduction - 40 Marks Viva-Voce - 10 Marks	70 Marks
	Total	**100 Marks reduced to 50 M**

Note:

- No Assignment and AAT
- **Minimum 40% passing marks in all divisions**

BNM Institute of Technology

Dept. of Computer Science and Engineering

Choice Based Credit System (CBCS and Outcome Based Education (OBE)

Semester: IV

Course Name: Python Programming and Applications		**Course Code: 22CSE145**
L: T: P: J	0:0:2:2	**CIA Marks:** 50
Credits:	**2**	**SEA Marks:** 50
Hours/Week (Total)	**4**	**SEA Duration:** 03 Hours

Course Learning Objectives: The students will be able to

1	Take a new computational problem and develop a plan to solve it through problem understanding and decomposition.
2	Follow a design creation process that includes specifications, algorithms, and testing.
3	Code, test, and debug a program in Python, based on your design.

Module-1: Introduction to Python	**No. of Hours**	**Blooms Cognitive Levels with CO mapping**
Introduction to Python: Entering Expressions into the Interactive Shell, The Integer, Floating-Point, and String Data Types, String Concatenation and Replication, Storing Values in Variables, Your First Program, Dissecting Your Program, Flow control: Boolean Values, Comparison Operators, Boolean Operators, Mixing Boolean and Comparison Operators, Elements of Flow Control, Program Execution, Flow Control Statements, Importing Modules, Ending a Program Early with sys.exit(), Functions: def Statements with Parameters, Return Values and return Statements, The None Value, Keyword Arguments and print(), Local and Global Scope, The global Statement, Exception Handling, A Short Program: Guess the Number Textbook 1: Chapter 1-3	**6**	**Apply CO1**

Sample Programs:

1. Develop a python code to check whether the given input is odd or even number.
2. Develop a python code to compare two number using nested conditionals.
3. Develop a python code to find out the largest of 3 numbers.
4. Develop a python code to find the factorial of a given number.
5. Develop a python code to generate the Fibonacci series up to n numbers.

Module-2: Data Structures		
Lists: The List Data Type, Working with Lists, Augmented Assignment Operators, Methods, Example Program: Magic 8 Ball with a List, List-like Types: Strings and Tuples, References, **Dictionaries and Structuring Data:** The Dictionary Data Type, Pretty Printing, Using Data Structures to Model Real-World Things, Textbook 1: Chapter 4-5	**6**	**Apply CO2**

Sample Programs:

1. Ask user to give name and marks of 10 different students. Store them in dictionary.
2. Take a list containg only strings. Now, take a string input from user and rearrange the elements of the list according to the number of occurence of the string taken from user in the elements of the list. E.g.-LIST : ["no bun","bug bun bug bun bug bug","bunny bug","buggy bug bug buggy"] STRING TAKEN : "bug" OUTPUT LIST:["bug bun bug bun bug bug","buggy bug bug buggy","bunny

bug","no bun"].
3. Count the number of occurrence of each letter in word "MISSISSIPPI". Store count of every letter with the letter in a dictionary.
4. Take 10 integer inputs from user and store them in a list. Again ask user to give a number. Now, tell user whether that number is present in list or not. (Iterate over list using while loop).

Module-3: Strings and Files

Manipulating Strings: Working with Strings, Useful String Methods, Project: Password Locker, Project: Adding Bullets to Wiki Markup. **Reading and Writing Files:** Files and File Paths, The os.path Module, The File Reading/Writing Process, Saving Variables with the shelve Module, Saving Variables with the print.format() Function, Project: Generating Random Quiz Files, Project: Multiclipboard. Textbook 1: Chapter 6, 8	**6**	**Apply CO3**

Sample Programs:
1. Develop a code to convert binary to decimal number, pass input as parameter to the function.
2. Develop a code to print calendar of a given month pass input as parameter to the function.
3. Develop a code to find the length of the string & "refrigerator" without using len function.
4. Write a program that takes your full name as input and displays the abbreviations of the first and middle names except the last name which is displayed as it is. For example, if your name is Pathireddy Santosh Reddy, then the output should be P.S.Reddy.
5. Develop a code to find the line starts with "T" from the file.
6. Write a program to read the contents of the file. If the file does not exist then raise appropriate exception.

Module-4: Classes & Objects

Classes and objects: Programmer-defined types, Attributes, Rectangles, Instances as return values, Objects are mutable, Copying. **Classes and functions:** Time, Pure functions, Modifiers, Prototyping versus planning. **Classes and methods:** Object-oriented features, Printing objects, Another example, A more complicated example, The __init__ method, The __str__ method, Operator overloading, Type-based dispatch, Polymorphism, Interface and implementation. Textbook 2: Chapter 15-17	**6**	**Apply CO4**

Sample Programs:
1. Develop a definition for a class named Circle with attributes center and radius, where center is a Point object and radius is a number. Instantiate a Circle object that represents a circle with its center at (150, 100) and radius 75.Write a function named point_in_circle that takes a Circle and a Point and returns True if the Point lies in or on the boundary of the circle.
2. Develop a python code to Calculate the Arc Length of an Angle by Assigning Values to the Radius and Angle Data Attributes of the class ArcLength.
3. Write a Program that prints the sum, difference and product of two complex numbers by creating a class named 'Complex' with separate functions for each operation where the real and imaginary parts are entered by user.

Module-5: NumPy and Pandas		
Introduction to NumPy: creating Array in NumPy, Accessing of Array Elements, NumPy Array Shape, Iterating Arrays, NumPy Built in Functions, NumPy ufuncs, Creating own ufunc, Arithmetic Functions, Rounding Decimals, Finding LCM and GCD. **Introduction to Pandas:** Series, Key/Value as Series, Data Frames, Loading a file into Data Frame, Viewing Data, Cleaning Data, Data Visualization using Matplotlib package. Link 1: https://www.w3schools.com/python/numpy/numpy_intro.asp Link 2: https://www.w3schools.com/python/default.asp	**6**	**Apply CO5**
Sample Programs: 1. Develop a code to create a series from a list, NumPy array and dictionary. 2. Develop a code for the following functions using pandas. 1.head() #Print starting 5 lines of information. 2.tail() #Print last 2 lines of information. 3.info() #Print the information. 3. Develop a code to multiply and add a 2D array. 4. Develop a python to plot a graphs(Scatter, Histogram, Bar, Pie) for any given dataset.		

Course Outcomes: After completing the course, the students will be able to	
22CSE145.1	Apply the Python Syntax and Semantics to understand the flow controls.
22CSE145.2	Develop python programs using core data structure.
22CSE145.3	Apply the concepts of Strings and file systems for problem solving.
22CSE145.4	Examine the OOP concepts for Application using python.
22CSE145.5	Interpret the NumPy and Pandas concepts for analysis and visualizations.

Textbooks

1. Al Sweigart, **"Automate the Boring Stuff with Python",** 1st Edition, No Starch Press, 2015. (Available under CC-BY-NC-SA license at https://automatetheboringstuff.com/)
2. Allen B. Downey, "Think Python: How to Think Like a Computer Scientist", 2 nd Edition, Green Tea Press, 2015. (Available under CC-BY-NC license at http://greenteapress.com/thinkpython2/thinkpython2.pdf

Reference Books

1. Gowrishankar S, Veena A, "Introduction to Python Programming", 1st Edition, CRC Press/Taylor & Francis, 2018. ISBN-13: 978-0815394372
2. Jake VanderPlas, "Python Data Science Handbook: Essential Tools for Working with Data", 1 st Edition, O'Reilly Media, 2016. ISBN-13: 978-1491912058
3. Charles Dierbach, "Introduction to Computer Science Using Python", 1 st Edition, Wiley India Pvt Ltd, 2015. ISBN-13: 978-8126556014
4. Wesley J Chun, "Core Python Applications Programming", 3rd Edition, Pearson Education India, 2015. ISBN-13: 978-9332555365

Marks Distribution for Assessment:

<table>
<tr><th>PBL</th><th>CIA</th><th>SEA</th><th colspan="3">CIA (50)</th><th>SEA
Conduction: 100 M
Reduced to: 50 M</th></tr>
<tr><td rowspan="4">Conduction</td><td rowspan="4">50</td><td rowspan="4">50</td><td rowspan="3">Theory</td><td>I IA</td><td>II IA</td><td rowspan="3">Project
Assessed for 100 marks
reduced to 50 Marks</td></tr>
<tr><td>25</td><td>25</td></tr>
<tr><td colspan="2">Average of 2 tests – 25 M</td></tr>
<tr><td>Practical</td><td colspan="2">Weekly Assessment
(Record/Project) – 10 Marks
Lab IA test – 15 Marks</td><td></td></tr>
<tr><td colspan="4">Total – 50 Marks</td><td>Total – 50 Marks</td></tr>
</table>

i) CIA: 50%

Theory - 2 IA tests - Each of 25 Marks	25 Marks
Practical **Weekly Assessment -** Lab record/Project – 10 Marks **Lab IA test –** 15 Marks	25 Marks
Total	**50 Marks**

ii) SEA : 50%

Project	Write up – 10 Marks Project report – 25 Marks Presentation & Demonstration - 50 Marks Viva-Voce – 15 Marks	100 Marks Reduced to 50 Marks
	Total	**50 Marks**

Note: Minimum 40% passing marks in all divisions

BNM Institute of Technology

Dept. of Computer Science and Engineering

Choice Based Credit System (CBCS and Outcome Based Education (OBE)

Semester: IV

Course Name: Web Technology and its Applications **Course Code: 22CSE146**

L: T: P: J	**0: 0: 2: 2**	**CIA Marks:** 50
Credits:	**2**	**SEA Marks:** 50
Hours/Week (Total)	**4 (30)**	**SEA Duration:** 03 Hours

Pre-Requisites: Fundamentals of Programming and Networking

Course Learning Objectives: The students will be able to

1	To explain web application development procedures
2	To impart servlet technology for writing business logic
3	To teach students the basics of server-side scripting using PHP
4	To facilitate students to connect to databases using JDBC

Module-1:	No. of Hours	BLL with CO mapping
Introduction to HTML: The development process, Html tags and simple HTML forms, web site structure Introduction to XHTML: XML, Move to XHTML, Meta tags, Character entities, frames and frame sets.	6	1 Understand
Module-2:		
Style sheets: Need for CSS, introduction to CSS, basic syntax and structure, using CSS, background images, colors and properties, manipulating texts, using fonts, borders and boxes, margins, padding lists, positioning using CSS, CSS2.	6	2 Understand
Module-3:		
JavaScript: Client-side scripting, what is JavaScript, how to develop JavaScript, simple JavaScript, variables, functions, conditions, loops and repetition DHTML: Combining HTML, CSS and JavaScript, events and buttons, controlling your browser	6	3 Apply
Module-4:		
XML: Introduction to XML, uses of XML, simple XML, XML key components, DTD and Schemas, Well formed, using XML with application.XML, XSL and XSLT. Introduction to XSL, XML transformed simple example, XSL elements.	6	4 Apply
Module-5:		
PHP: Introduction and basic syntax of PHP, decision and looping with examples, PHP and HTML, Arrays, Functions, Browser control and detection, string, Form processing, Files, Advance Features: Cookies and Sessions. Databases: Basic command with PHP examples, Connection to server, creating database, selecting a database, listing database, listing table names creating a table, inserting data, altering tables, queries, deleting database, deleting data and tables.	6	5 Create

Handson Practice sets	
Practical Set -1 HTML	
✓	Design web pages for your college containing a description of the courses, departments, faculties, library etc, use href, list tags.
✓	Create your class timetable using table tag.
✓	Create user Student feedback form (Use textbox, text area, checkbox, radio button, select box etc.)

✓	Create your resume using HTML tags also experiment with colors, text, link, size and also other tags you studied.
✓	At the bottom create a link to take user to the top of the page
Practical Set -2 CSS	
✓	Design a web page of your home town with an attractive background color, text color, an Image, font etc. (use internal CSS).
✓	Use Inline CSS to format your resume that you created.
✓	Use External CSS to format your class timetable as you created.
✓	Use External, Internal, and Inline CSS to format college web page that you created.
Practical Set -3 JavaScript	
✓	Develop a JavaScript to display today's date.
✓	Develop simple calculator for addition, subtraction, multiplication and division operation using JavaScript
✓	Create HTML Page with JavaScript which takes Integer number as input and tells whether the number is ODD or EVEN.
✓	Create HTML Page that contains form with fields Name, Email, Mobile No,Gender, Favorite Color and a button now write a JavaScript code to combine and display the information in textbox when the button is clicked.
✓	Implement Validation in above Feedback Form.
✓	Use regular expression for validation in Feedback Form.
✓	Write a JavaScript program to change background color after 5 seconds of page load.
Practical Set -4 XML	
✓	Create XML file to store student information like Enrollment Number, Name, Mobile Number, Email Id.
Practical Set -5 PHP	
✓	Write a php program to display today's date in dd-mm-yyyy format.
✓	Write a php program to check if number is prime or not.
✓	Create HTML page that contain textbox, submit / reset button. Write php program to display this information and also store into text file.
✓	Write a PHP Script for login authentication. Design an html form which takes username and password from user and validate against stored username and password in file.
✓	Write PHP Script for storing and retrieving user information from MySql table. 1. Design A HTML page which takes Name, Address, Email and Mobile No. From user (register.php) 2. Store this data in Mysql database / text file. 3. Next page display all user in html table using PHP (display.php)
✓	Write a PHP script for user authentication using PHP-MYSQL. Use session for storing username.
Mini Project: Website	
Students have to create a website which contains above topics in Website.	

Course Outcomes: After completing the course, the students will be able to	
22CSE146 .1	Understand the HTML tags and use them to develop the user-friendly web pages.
22CSE146 .2	Understand the CSS with its types and use them to provide the styles to the web pages at various levels
22CSE146 .3	Develop the dynamic web pages by using the JavaScript
21CSE146 .4	Build the web pages dynamically using the database connectivity and applying server-side scripting with XML and PHP
21CSE146 .5	Create the modern Web applications using the client and server-side technologies and the web design fundamentals.

Reference Books
• Developing Web Applications, Ralph Moseley and M. T. Savaliya, Wiley-India • Web Technologies, Black Book, dreamtech Press • HTML 5, Black Book, dreamtech Press • Developing Web Applications in PHP and AJAX, Harwani, McGrawHil

Marks Distribution for Assessment:

<table>
<tr><th>PBL</th><th>CIA</th><th>SEA</th><th colspan="3">CIA (50)</th><th>SEA
Conduction: 100 M
Reduced to: 50 M</th></tr>
<tr><td rowspan="4">Conduction</td><td rowspan="4">50</td><td rowspan="4">50</td><td rowspan="3">Theory</td><td>I IA</td><td>II IA</td><td rowspan="4">Project
Assessed for 100 marks
reduced to 50 Marks</td></tr>
<tr><td>25</td><td>25</td></tr>
<tr><td colspan="2">Average of 2 tests – 25 M</td></tr>
<tr><td>Practical</td><td colspan="2">Weekly Assessment
(Record/Project) – 10 Marks
Lab IA test – 15 Marks</td></tr>
<tr><td colspan="4">Total – 50 Marks</td><td>Total – 50 Marks</td></tr>
</table>

i) CIA: 50%

Theory - 2 IA tests - Each of 25 Marks	25 Marks
Practical **Weekly Assessment -** Lab record/Project – 10 Marks **Lab IA test –** 15 Marks	25 Marks
Total	**50 arks**

ii) SEA : 50%

Project	Write up – 10 Marks Project report – 25 Marks Presentation & Demonstration - 50 Marks Viva-Voce – 15 Marks	100 Marks Reduced to 50 Marks
	Total	**50 arks**

Note: Minimum 40% passing marks in all divisions

B.N.M. Institute of Technology

An Autonomous Institution under VTU

Semester: IV		
COURSE: CONSTITUTION OF INDIA AND PROFESSIONAL ETHICS		
Course Code: 22CIP147	**L:T:P:J: 0:2:0:0**	**CIA Marks:** 100
Credits:	**1**	**SEA Marks:**
Hours:	**15 hrs**	

Course Learning Objectives: The students will be able to

1	know the fundamental political codes, structure, procedures, powers, and duties of Indian government institutions, fundamental rights, directive principles, and the duties of citizens
2	know the Indian top civil service positions and the exams conducted by UPSC and SPSC for the same
3	Understand engineering ethics and their responsibilities; identify their individual roles and ethical responsibilities towards society.

MODULE 1: Introduction to Indian Constitution	**RBT**	**Hrs**
The Necessity of the Constitution, Introduction to Indian Constitution, The Making of the Constitution, Role of Constituent Assembly, Preamble and Salient features of the Constitution of India, Fundamental Rights and its Restriction and limitations in different complex situations, Directive Principles of State Policy, Fundamental Duties.	**1,2,3**	**3**
MODULE 2: System of Government, Central Government, State Government	**RBT**	**Hrs**
System of Government-Parliamentary System, Federal System. Central Government-Basic details, Powers and Functions of Union Executive. Parliament- LS and RS (Composition, Duration, Membership and Presiding officers of Parliament and their functions). Leaders in Parliament (Leader of the House and Leader of the Opposition). Sessions of Parliament (Summoning, Adjournment, Adjournment Sine Die, Prorogation, Dissolution). Quorum of House, Language in Parliament, Joint sitting of two Houses. State Government-Basic details, Powers and Functions of State Executive. State Legislature (Composition, Duration, Membership and Presiding officers of Parliament and their functions).	**1,2,3**	**3**
MODULE 3: Judiciary, Amendments and Emergency Provisions	**RBT**	**Hrs**
Supreme Court, High Court, Judicial Review, Judicial Activism. Methods in Constitutional Amendments (How and Why). Types of Emergencies and its Consequences, Recent Amendments to the Constitution.	**1,2,3**	**3**
MODULE 4: Elections, Constitutional and Non Constitutional Bodies	**RBT**	**Hrs**
Elections- Election Commission of India, Electoral Process. Constitutional Bodies- Election Commission, Union Public Service Commission, State Public Service Commission, Goods and Service Tax Council. Non Constitutional Bodies- Central Information Commission, State Information Commission.	**1,2,3**	**3**

MODULE 5: Professional Ethics	RBT	Hrs
Scope & Aims of Engineering & Professional Ethics, Positive and Negative Faces of Engineering Ethics, Responsibilities in Engineering, the impediments to Responsibility. Trust and Reliability in Engineering, Risks, Safety and liability in Engineering, Clash of Ethics, IPRs (Intellectual Property Rights)	**1,2,3**	**3**

Course outcome: On completion of this course, students will be able to,
CO1: Have constitutional knowledge and legal literacy.
CO2: Have knowledge on All India Services and State Civil Services.
CO3: Understand Engineering and Professional Ethics and responsibilities of Engineers.

Reference Books

Suggested Learning Resources:

1. **Title of the Book - Indian Polity**
   Name of the Author - M Lakshmikanth
   Name of the Publisher-Mc Graw Hill Education
   Edition and Year- 2019
2. **Title of the Book - Engineering Ethics**
   Name of the Authors - M. Govindarajan, S.Natarajan, V.S. Senthilkumar
   Name of the Publisher- Prentice-Hall
   Edition and Year-2004

3. Durga Das Basu (DD Basu): "Introduction to the Constitution on India", (Students Edition.) Prentice –Hall EEE, 19th / 20th Edn., (Latest Edition) or 2008.

4. Shubham Singles, Charles E. Haries, and Et al : "Constitution of India and Professional Ethics" byCengage Learning India Private Limited, Latest Edition – 2018.

5. M.Govindarajan, S.Natarajan, V.S.Senthilkumar, "Engineering Ethics", Prentice –Hall of India Pvt. Ltd. New Delhi, 2004

6. M.V.Pylee, "An Introduction to Constitution of India", Vikas Publishing, 2002.

7. Latest Publications of **NHRC - Indian Institute of Human Rights**, New Delhi.

Web Links and Video Lectures

www.unacademy.com/lesson/future-perfect-tense/YQ9NSNQZ https://successesacademy

BNM Institute of Technology

Syllabus for Softskills-2

SEMESTER – IV

Subject Name	Softskills-2 (Aptitude Quantitative & Logical)	**Weekly Assessment Marks**	10
Subject Code	22SFT148	**Internal Assessment Marks**	60
Number of Contact Hours/Week	2	**Company Simulation Tests Marks**	30
Total Number of Contact Hours	36	**Credits**	1
Module 1 (Quantitative Aptitude - 1)	**Number System -** Classification of Numbers, Multiple and factors, Divisibility Rules		
	HCF & LCM, Squares and Cubes.		
	Profit & Loss - Concepts of SP, CP, Profit, Loss, Gain or Loss %.		
	Profit & Loss - Marked Price & Discount problems, Successive Discount.		
	Percentages – Percent To Decimal Or Fraction Conversion, Inverse Case – Value From Percentage, relative Percentage		
	Averages - Understanding Averages & solving problems.		
Module 2 (Quantitative Aptitude - 2)	**Ratios -** Duplicate and Triplicate Ratio, Direct and Indirect variation		
	Proportion - Direct Indirect proportion and relation.		
	Simple Interest - Simple Interest, Basic Difference b/w both the Interests		
	Compound Interest - CI with a Fractional Rate, to find Instalments.		

	Speed Time & Distance - Important formulas, Relative Speed.
	Speed Time & Distance - Understanding Units & Conversion of units
Module 3 (Quantitative Aptitude - 3)	**Time & Work -** Introduction and Concept, Important Time and Work Formula, Work Done
	Time & Work - Rate of Work, Time Taken, If a piece of work is done in x number of days
	Data Interpretation - Bar Graph, Tabular Form, Line Chart, case let Form
	Data Interpretation - Pie Chart, Radar/Web, and Missing Data Interpretation.
	Probability – Understanding concepts and important formulas.
	Probability – Understanding types of problems on probability
Module 4 (Logical - 1)	Problems on Syllogisms
	Problems on Assumptions
	Logical Puzzles - K-level thinking
	Logical Puzzles - Arithmetic Puzzles
	Stick Puzzles
	Series Completion - Basics of Next no, Missing no and Wrong no and problems on that.
	Solving various types of Letter series and understanding different types.
Module 5 (Logical - 2)	**Problem on Ages -** Understanding concepts and basic formula along with solving different types of problems.
	Problem on Ages - Tips and Tricks to Solve Problems on Ages

	Blood Relation - Generation Tree, Family Tree Problems.
	Blood Relation - Statement Based Questions, Coded Blood Relation Question.
	Coding & Decoding - Concept of EJOTY, Opposite Letter, Reversing the alphabets.
	Coding & Decoding - Jumbling of Letter, Finding Codes of Derivatives.
Module 6 (Logical - 3)	**Clocks –** Understanding concepts and basic formula along with solving different types of problems.
	Calendar - Understanding concepts and basic formula along with solving different types of problems.
	Image Analysis - Paper cutting & Folding, Mirror & Water Image, Cubes and Dice, Analogy, Find the odd one out, Rule Detection
	Odd Man Out - Following certain patterns and groups.
	Identifying the errors/odd one in the group.
	Seating Arrangement - Linear and Circular seating Arrangements as well as problems of sitting around Square and Rectangular.
	Distance & Direction - Distance and Displacement between any two points as well as puzzles based on that, Concept of Shadows.