THE NATIONAL INSTITUTE OF ENGINEERING

MYSORE – 8

(Autonomous Institution under VTU)

B.E – CSE (AIML)

Scheme of III – IV Semester

Department of Computer Science and Engineering

The National Institute of Engineering, Mysuru														
Scheme of Teaching and Examination														
Department: Computer Science and Engineering														
B.E. in CSE - 2024 Admitted Batch														
Semester : III														
Sl. No.	**Type of Course**	**Course Code**	**Course Title**	**Teaching Department (TD)**	**Question Paper setting Board (PSB)**	**Teaching Hrs/Week**				**Examination**				
						L	**T**	**P**	**S**	**Duration in Hours**	**CIE Marks**	**SEE Marks**	**Total Marks**	
1	PCC/BSC	BCS301	Mathematics for Computer Science	TD: Maths	PSB: Maths	3	2	0	-	3	50	50	100	**4**
2	IPCC	BCS302	Digital Design & Computer Organization	TD: CS	PSB: CS	3	0	2	-	3	50	50	100	**4**
3	IPCC	BCS303	Operating Systems	TD: CS	PSB: CS	3	0	2	-	3	50	50	100	**4**
4	PCC	BCS304	Data Structures and Applications	TD: CS	PSB: CS	3	0	0	-	3	50	50	100	**3**
5	PCCL	BCSL305	Data Structures Laboratory	TD: CS	PSB: CS	0	0	2	-	3	50	50	100	**1**
6	ESC	BXX306X	ESC/ETC/PLC	TD: CS	PSB: CS	3	0	0	-	3	50	50	100	**3**
						OR								
						2	0	2	-					
7	UHV	BSCK307	Social Connect and Responsibility	TD: CS	PSB: CS	0	0	2	-	1	100	-	100	**1**
8	AEC/SEC	BXX358X	Ability Enhancement Course (AEC)	TD: CS	PSB: CS	If the course is a Theory					50	-	100	**1**
						1	0	0	-	1				
						If the course is a Laboratory								
						0	0	2	-	2				
9	MC	BNSK359	National Service Scheme (NSS)	NSS Coordinator		0	0	2	-	-	100	-	100	**0**
		BPEK359	Physical Education (PE) (Sports & Athletics)	PE D										
		BYOK359	Yoga	Yoga Teacher										
		BMUK359	Music	Music Teacher										
Total											**550**	**300**	**900**	**21**

PCC: Professional Core Course, PCCL: Professional Core Course laboratory, UHV: Universal Human Value Course, MC: Mandatory Course (Non-credit), AEC: Ability Enhancement Course, SEC: Skill Enhancement Course, L: Lecture, T: Tutorial, P: Practical: Skill Development Activity, CIE: Continuous Internal Evaluation, SEE: Semester End Evaluation. K :This letter in the course code indicates common to all the stream of engineering. ESC: Engineering Science Course, ETC: Emerging Technology Course, PLC: Programming Language Course

Engineering Science Course (ESC/ETC/PLC)			
BCS306A	Object Oriented Programming with Java	BCS306D	JavaScript
BCS306B	Object Oriented Programming with C++		
Ability Enhancement Course -IV			
BCS358A	Data Analytics with Excel	BCS358D	Data Visualization with Python
BCS358B	Data Analytics with R		
BCS358C	Project Management with Git		

Code: BCS302 **Course: Digital Design & Computer Organization**

Credits: 4 **CIE: 50 Marks**

L:T:P - 3:0:2 **SEE: 50 Marks**

SEE Hours: 3 **Total Marks:100**

Prerequisites if any	**Fundamentals of Logic**
Learning objectives	• To provide the knowledge to explain the fundamentals of Logic circuits and combinational circuits. • To introduce the design of sequential circuit systems, Registers and Counters • The basics involved in number representation and arithmetic operations in the computer system. • Basic processor concept, instruction execution and Bus architecture, Memory architecture and mapping techniques.

Course Outcomes:

On the successful completion of the course, the student will be able to

COs	**Course Outcomes**	**Bloom's level**
CO1	Understand combinational and sequential circuits, memory and control architecture, and algorithms for arithmetic operations and floating-point representation.	Understand
CO2	Apply techniques of logic design to circuits, memory architecture, and arithmetic algorithms.	Apply
CO3	Analyze logic circuits, registers and counters, memory architecture with control systems and algorithms for efficient arithmetic.	Analyze
CO4	Evaluate digital circuits including combinational, sequential, registers, counters, memory, control systems, and arithmetic algorithms with floating-point representation.	Evaluate

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12		PSO1	PSO2
CO1	2	3	3	3	3	1	1	2	2	-	-			3	2
CO2	3	3	3	3	3	2	-	2	2	1	1			3	2
CO3	3	3	3	3	2	2	1	-	2	1	-			2	2
CO4	3	3	2	3	3	2	-	1	2	1	-			2	-

Mapping Strength: Strong– 3 Medium – 2 Low – 1

Course Structure

Sl. No.	Modules	No. of Lecture Hour s	No. of Tutorial Hours	No. of Practical Hours
	Module 1--Combinational Circuits			
1.1	**Introduction to Boolean functions:** Basics of Logic Gates, Boolean Algebra, Boolean Theorems, Basic gates and Universal Gate **Simplification of Boolean Functions**: The Map Method, Two-and Three Variable Maps, Four- Variable Maps, Don't-Care Conditions	2	-	-
1.2	**Combinational Logic**: Adders, Subtractors	3	-	2
1.3	BCD to Excess-3 code converter	2	-	1
1.4	Decoders, Multiplexer	3	-	2
	Module 2-Sequential Circuits			
2.1	**Sequential Logic**: Introduction, Flip-Flops, Triggering of Flip Flops	2	-	2
2.2	Analysis of Clocked Sequential Circuits	2	-	-
2.3	State Reduction and Assignment	1	-	-
2.4	Design Procedure	1	-	-
2.5	Registers, Counters: Introduction to Registers	1	-	-
2.6	Shift Registers	2	-	1
2.7	Synchronous & Asynchronous Counters.	1	-	2
	Module 3-Processing Unit			
3.1	**Introduction**: Introduction to Computer Organization, Basic Concepts, Functional Units, Basic Operational Concepts	1	-	-
3.2	**The Processor**: Fundamental concepts, Performance Equation **Machine Instructions and Programs**: Memory Location and Addresses, Memory Operations, Instruction and Instruction sequencing, Addressing Modes.	2	-	-
3.3	Execution of complete instruction, Multiple-Bus organization	2		
3.4	Hardwired control unit and Micro programmed control	2	-	-

Module 4-Arithmetic Unit				
4.1	**Arithmetic unit:** Multiplication of Positive numbers A signed operand multiplication	2	-	-
4.2	Bit pair recoding of multipliers, carry save addition of summands	3	-	-
4.3	Integer division, Restoring Algorithm and Non-Restoring Algorithm	2	-	-
Module 5-Memory Unit				
5.1	**Memory Unit:** Basic concepts, Internal organization of memory chips, Structure of larger memory.	3	-	-
5.2	Cache memories, Mapping functions	3	-	-
Total No. of Lecture Hour		**40**	-	-
Total No. of Tutorial Hours			**00**	-
Total No. of Practical Hours				**10**

Integrated Lab Component: Digital Design & Computer Organization		
Sl. No	**COs**	**PART-A HARDWARE**
1.	CO1	Implement Half adder and Full adder using logic gates. Implement Half subtractor, Full subtractor using logic gates
2.	CO1	Implement BCD to Excess-3 using basic gates.
3.	CO1	Realize a full adder or any Boolean function using 3:8 decoder IC.
4.	CO1	Given any four variable logic expression, realize the simplified logic expression using 8:1 multiplexer
5.	CO2	Realize a JK master Slave Flip- Flop and verify its truth table.
6.	CO2	Realize mod-4 and mod-6 counter using Synchronous counter design
		PART-B SIMULATION USING XILINX
1.	CO1	Design Verilog HDL to implement simple circuits using structural, Data flow and Behavioural model.
2.	CO1	Write a VHDL code for a Half –Adder , Simulate and verify its working.
3.	CO1	Write a VHDL code for a Full –Adder , Simulate and verify its working.
4.	CO1	Write a VHDL code for a Full –Subtractor, Simulate and verify its working.
5.	CO1	Write a VHDL code for 2:1, 4:1 and 8:1 multiplexer , Simulate and verify its working.
6.	CO2	Write a VHDL code for a SR, D and JK flip flop , Simulate and verify its working.

Text Books:

- Digital Design: M. Morris Mano, Pearson (2013), new print 5th edition
- Computer Organization: C Hamacher, Z Vranesic, S Zaky:, Tata McGraw Hill, 5thEdition, 2011.

Reference Books:

- Computer Architecture A Quantitative Approach: John L Hennessy, David A Patterson,Elsevier, 5th Edition 2012.
- The Elements of Computing System – Building the Modern Computer from FirstPrinciples: Noam Nisan, Shimon Schocken, The MIT Press (2005).
- Digital Principles and Applications: Donald P Leach, Albert Paul Malvino & Goutham Saha,TMH, 6th Edition, 2006.

Online Resources:

NPTEL: Switching Circuits and logic design

https://onlinecourses.nptel.a c.in/noc21_cs64

NPTEL: Computer Organization and Architecture A Pedagogical Aspect

https://onlinecourses.nptel.ac.in/noc19_cs04/preview

Edx: Computation Structures 3:Computer Organization

https://www.edx.org/course/computation-structures3-cmputer-mitx-6-004-3x0

Coursera: Digital Systems - https://www.coursera.org/learn/digital-systes

Code: BCS303 **Course: Operating Systems**

Credits: **CIE: 50 Marks**

L:T:P - 3:0:2 **SEE: 50 Marks**

SEE Hours: 3 **Total Marks: 100**

Prerequisites if any	Basic knowledge of Computer Organization and Programming in C/C++
Learning objectives	1. To provide a foundation on the structure and functioning of operating systems with emphasis on processes, threads, and scheduling. 2. To develop an understanding of process synchronization and interprocess communication mechanisms with exposure to classical problems. 3. To introduce concepts of memory management and memory allocation strategies in operating systems. 4. To familiarize students with page replacement policies and file system design issues. 5. To impart knowledge of deadlock concepts and methods for its prevention, avoidance, detection, and recovery.

Course Outcomes:

On the successful completion of the course, the student will be able to

COs	Course Outcomes	Bloom's level
CO1	Describe core Operating systems concepts: processes, threads, scheduling, IPC, memory, file systems, and deadlocks.	Understand
CO2	Implement scheduling, synchronization, memory management, page replacement, file operations, and deadlock handling.	Apply
CO3	Examine Operating systems behavior and performance in scheduling, memory, IPC, and deadlock scenarios.	Analyze
CO4	Assess Operating systems algorithms for scheduling, memory, files, and deadlocks to recommend efficient designs.	Evaluate

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12		PSO1	PSO2
CO1	**3**	**2**	**-**	**2**	**2**	**-**	**-**	**-**	**-**	**1**	**-**	**1**		**3**	**2**
CO2	**3**	**3**	**2**	**3**	**3**	**-**	**-**	**-**	**-**	**-**	**1**	**1**		**3**	**3**
CO3	**3**	**2**	**3**	**3**	**2**	**1**	**-**	**-**	**1**	**-**	**1**	**1**		**3**	**3**
CO4	**3**	**2**	**3**	**2**	**3**	**1**	**1**	**-**	**1**	**1**	**1**	**1**		**3**	**3**

Mapping Strength: Strong– 3 Medium – 2 Low – 1

Sl. No.	Modules	No. of Lecture Hours	No. of Tutorial Hours	No. of Practical Hours
Module – 1: Processes, Threads and Scheduling				
1.1	**Introduction:** what is an operating system? The Operating system as an extended machine, The Operating system as a resource manager	1	-	-
1.2	System calls for process Management	1	-	1
1.3	**Processes**: The process model, Process creation, Process termination	1	-	-
1.4	Process hierarchies, Process states.	1	-	-
1.5	**Threads**: Thread Usage, Thread Model	1	-	-
1.6	Implementing threads in user space, implementing threads in the kernel. Hybrid implementations	1	-	-
1.7	**Scheduling**: Introduction to scheduling, Scheduling in Batch Systems (FCFS, SJF, SRTN)	2	-	2
1.8	Scheduling in interactive systems: Priority based Scheduling, Round robin Scheduling	1	-	1
1.9	Guaranteed, Lottery and Fair Share Scheduling	1	-	-
Module – 2: Interprocess Communication				
2.1	**Interprocess communication:** Race conditions, Critical regions	1	-	-
2.2	Mutual exclusion with busy waiting	1	-	-
2.3	Sleep and wakeup	1	-	-
2.4	Semaphores	1	-	1
2.5	Mutexes, Message Passing	1	-	-
2.6	**Classical IPC problems:** The dining philosophers' problem	1	-	-
2.7	The Readers and Writers problem. Case Study: D-Bus	1	-	-
Module – 3: Memory Management				

3.1	**Memory Management:** Background	1	-	-
3.2	Swapping	1	-	-
3.3	Contiguous Memory Allocation	1	-	1
3.4	Paging	1	-	-
3.5	Structure of the page Table	1	-	-
3.6	Segmentation.	1	-	-
3.7	**Virtual Memory**: Background; Demand Paging, copy-on write	1	-	-
3.8	Thrashing	1	-	-
Module – 4: Page replacement Algorithms and Design Issues				
4.1	**Page Replacement Algorithms**: The optimal page replacement algorithm	1	-	-
4.2	The not recently used page replacement algorithm	1	-	-
4.3	The first-in first-out.	1	-	1
4.4	The second chance page replacement algorithm	1	-	-
4.5	The clock page replacement algorithm, The least recently used page replacement Algorithm.	1	-	1
4.6	**Design issues for paging systems**: Local versus Global allocation policies, Load control, Page size	1	-	
4.7	**File system implementation**: File System Layout, Implementing Files	1	-	-
4.8	Implementing Directories.	1	-	-
Module – 5: Deadlocks				
5.1	**Deadlocks**: **Introduction to deadlocks**: Conditions for deadlock, Deadlock modeling.	1	-	-
5.2	**Deadlock detection and recovery**: Deadlock detection with one resource of each type,	1	-	-

5.3	Deadlock detection with multiple resource of each type, Recovery from deadlock.	1	-	1
5.4	**Deadlock avoidance**: Resource trajectories, Safe and Unsafe States, The Banker's algorithm for a single resource,	1	-	-
5.5	The banker's algorithm for multiple resources.	1	-	1
5.6	**Deadlock prevention**: Attacking the Mutual-Exclusion Condition, Attacking the Hold-and-Wait condition, Attacking the No-Preemption condition, Attacking the Circular Wait condition	1	-	-
5.7	I**ssues**: Two phase locking, Starvation	1		
	Total No. of Lecture Hours	**40**	-	-
			Total No. of Tutorial Hours	**00**	**-**
				Total No. of Practical Hours	**10**

List of Lab Programs

Sl. No.	Lab Programs
1	Implement the process system calls fork (), exec () and wait ()
2	a) Implement FCFS (First Come First Serve) scheduling algorithm b) Implement SJF (Shortest Job First) scheduling algorithm.
3	Implement Priority Scheduling algorithm.
4	Implement Round Robin Scheduling algorithm.
5	Write a program to implement Producer-Consumer Problem using semaphores.
6	Implement Memory Allocation methods for Fixed partition using First Fit, Worst Fit, Best Fit
7	Write a program to implement FIFO page replacement algorithm.
8	Write a program to implement LRU page replacement algorithm.
9	Write a program to detect Deadlock.
10	Write a program to implement Banker's algorithm for Deadlock avoidance.

Text Books:

1. Modern Operating systems, 4th Edition, Andrew S.Tanenbaum, Herbert Bos, Pearson Education Limited; Global Edition. (MODULE1, MODULE 2, MODULE 4, MODULE 5)
2. Operating System Concepts – Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, Wiley- India, 8th edition (MODULE 3)

Note: Refer Text book no.2 for problems on Process scheduling and Page Replacement Algorithms

Reference Books:

1. Operating Systems, William Stallings, PHI, Fourth Edition
2. Operating Systems, Milan Kovic, Tata McGraw Hill, 2001.
3. Operating System Design: v. 1: The Xinu Approach (Prentice-hall Software) Douglas Comer.
4. TUNIX Operating Systems: A Practical Approach, Robert Switzer, Prentice-Hall, 1993.

Online Resources: http://www.nptel.ac.in/courses/106108101/

Course Code: BCS304 **Course: Data Structures and Applications**

Credits: 3 **CIE: 50 Marks**

L:T:P: 3:0:0 **SEE: 50 Marks**

SEE Hours: 3 **Total Marks: 100**

Prerequisites if any	Problem Solving through Programming
Learning objectives	• Understand fundamental linear and non-linear data structures and their role in problem-solving. • Apply data structures to implement efficient solutions for computational problems.

Course Outcomes:

On the successful completion of the course, the student will be able to

COs	Course Outcomes	Bloom's level
CO1	Explain the concepts of pointers, structures, and the classification of data structures.	Understand
CO2	Apply linked lists, stacks, and queues to solve computational problems.	Apply
CO3	Analyze the operations and applications of tree structures, sorting methods, and hashing techniques for solving computational problems.	Analyze
CO4	Design and evaluate solutions to real-world problems using appropriate data structures	Evaluate

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PO 4	PO5	PO6	PO7	PO 8	PO9	PO1 0	PO1 1	PO1 2		PSO 1	PSO 2
CO1	3	2	2	-	2	-	-	-	1	-	-	2		-	-
CO2	3	2	3	2	3	-	-	-	1	1	-	2		2	2
CO3	3	3	3	3	3	-	-	-	1	1	-	2		3	3
CO4	3	3	3	3	3	-	-	-	2	3	3	3		3	3

Mapping Strength: 3 – Strong 2 – Medium 1 – Low

Course Structure

Nos	Modules	No. of Lecture Hours	No. of Tutorial Hours	No. of Practical Hours
Module – 1 Pointers and Data Structures Fundamentals				
1.1	**Pointers:** Introduction	1	-	-
1.2	Structures, Nested Structures, Arrays of Structures	2	-	-
1.3	Structures and Functions, Self-Referential Structures	2	-	-
1.4	**Introduction to Data Structures,** Classification of data structures, Operations on data structures, Abstract data type	1	-	-
Module – 2 Linked Lists				
2.1	**Linked Lists -** Singly linked lists	3	-	-
2.2	Circular linked lists	2	-	-
2.3	Doubly linked lists	2	-	-
2.4	Circular doubly linked lists	2	-	-
Module – 3 Stacks and Queues				
3.1	**Stacks**: Introduction, Array Representation of Stacks	1	-	-
3.2	Operations on a Stack, Linked Representation of Stacks	1	-	-
3.3	**Applications of Stacks:** Conversion of an infix expression into a postfix expression, Evaluation of a postfix expression, Recursion.	3	-	-
3.4	**Queues:** Introduction, Array Representation of Queues, Linked Representation of Queues	2	-	-
3.5	**Types Of Queues:** Circular Queue, Priority Queues	2	-	-
Module – 4 Trees				
4.1	**Trees:** Introduction, Types of Trees	1	-	-
4.2	**Traversing** A Binary Tree,	1	-	-
4.3	Binary Search Trees, Operations on Binary Search Trees	3	-	-
4.4	**Threaded Binary Trees:** One-way Threading,	1	-	-
4.5	AVL Tree, Binary Heaps operations	3	-	-
Module – 5 Sorting and Hashing				
5.1	**Sorting:** Introduction to Sorting, Radix Sort, Heap Sort	2	-	-
5.2	**Hashing and Collision:** Introduction, Hash Tables, Different Hash Functions, Collisions, Pros and Cons of Hashing, Applications of Hashing	5	-	-
	Total No. of Lecture Hours	**40**	-	-
		Total No. of Tutorial Hours	**0**	-
			Total No. of Practical Hours	**0**

Text Books:

1. Data Structures using C, Reema Thareja, 2nd Edition, 2018, Oxford University Press.
2. Data Structures using C, Aaron M Tenenbaum, Yedidyah Langsam and Moshe J Augenstein, 2014, low price edition, Pearson education.

Reference Books:

1. Richar F Gilberg and Behronz A Forouzan, ―Data Structures, A Pseudocode Approach with C, 2nd Edition, 2012, Thomson.
2. Horowitz, Sahni, Anderson-Freed,―Fundamentals of Data Structures in C, 2nd Edition, 2011, Universities Press.

Online Resources:

1. NPTEL: Data Structures and Algorithms Design - https://onlinecourses.nptel.ac.in/noc25_cs81/preview
2. Coursera: Data Structures - https://www.coursera.org/programs/faculty-learning-program-iqr5x/learn/data-structures

Course Code: BCSL305 **Course: Data Structures Laboratory**

Credits: 1 **CIE: 50 Marks**

L:T:P: 0:0:2 **SEE: 50 Marks**

SEE Hours: 2 **Total Marks: 100**

Prerequisites if any	Fundamentals of C programming
Learning objectives	• Apply programming constructs and data structure operations to implement problem-solving solutions. • Analyze and compare different data structures through hands-on experiments in varied problem contexts.

Course Outcomes:*On the successful completion of the course, the student will be able to*

COs	Course Outcomes	Bloom's level
CO1	Implement concepts of pointers, dynamic memory allocation, and linked lists to solve programming problems.	Apply
CO2	Apply stack, queue, tree, sorting, and hashing techniques for problem-solving in various contexts.	Apply
CO3	Apply tree structures, sorting algorithms, and hashing techniques in program development.	Apply

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12		PSO1	PSO2
CO1	3	2	3	2	2	-	-	-	1	-	-	2		3	2
CO2	3	3	3	2	3	-	-	-	1	-	-	2		2	2
CO3	3	3	3	3	3	-	-	-	1	1	-	2		3	3

Mapping Strength : 3 – Strong 2 – Medium 1 – Low

List of Experiments

Sl. No.	Experiment	CO Mapped
1	(a) Write a program to accept 3 integers and determine the maximum using functions and pointers. (b) Write a program to search for a given element in an array using pointers. (c) Write a program to find the maximum element in each row of a matrix using pointers.	CO1
2	(a) Define a structure TIME with hour, minute, second, and a self-referencing pointer. Dynamically create two variables, link them, and display in h:m:s format. (b) Define a structure Student with members name, USN, and marks in 3 subjects. Store *n* student records, compute averages, and display in descending order of marks.	CO1
3	(a) Implement a singly linked list with operations: insert at front, insert at position, delete by value, search, and display. (b) Extend the program to reverse the linked list elements. (c) Implement a circular singly linked list with operations: insert at end, delete by key, and display nodes.	CO2
4	(a) Implement a doubly linked list with operations: insert at front, insert to the left of a key node, delete by value, and display. (b) Implement a circular doubly linked list with operations: insert at front/end, delete by key, and display nodes in forward and reverse directions.	CO2
5	(a) Implement stack operations (push, pop, display, overflow/ underflow) using arrays. (b) Convert an infix expression to postfix using a stack. (c) Evaluate a postfix expression. (d) Check palindrome using a stack. (e) Use recursion to compute the nth Fibonacci number.	CO2
6	(a) Implement a queue using arrays with operations: insert, delete, and display (size = 5). (b) Implement a circular queue using linked lists.	CO2
7	(a) Construct a Binary Search Tree (BST) and perform inorder, preorder, and postorder traversals. (b) Count the number of leaf nodes in the BST. (c) Print all root-to-leaf paths in the BST.	CO3
8	(a) Implement radix sort on a set of integers. (b) Implement hashing with collision handling (closed hashing). Support insert and search operations.	CO3

Text Books:

1. Data Structures using C, Reema Thareja, 2nd Edition, 2018, Oxford University Press.
2. Data Structures using C, Aaron M Tenenbaum, Yedidyah Langsam and Moshe J Augenstein, 2014, low price edition, Pearson education.

Reference Books:

1. Richar F Gilberg and Behronz A Forouzan, "Data Structures, A Pseudocode Approach with C", 2nd Edition,2012, Thomson.
2. Horowitz, Sahni, Anderson-Freed, "Fundamentals of Data Structures in C", 2nd Edition 2011, Universities Press.

Online Resources:

1. NPTEL: Data Structures and Algorithms Design - https://onlinecourses.nptel.ac.in/noc25_cs81/preview
2. Coursera: Data Structures - https://www.coursera.org/programs/faculty-learning-program-iqr5x/learn/data-structures

ENGINEERING SCIENCE COURSE (ESC/ ETC/ PLC)

Code: BCS306A **Course: Object Oriented Programming with Java**

Credits: 3 **CIE: 50 Marks**

L:T:P - 2:0:2 **SEE: 50 Marks**

SEE Hours: 3 **Total Marks: 100**

Prerequisites if any	**C programming**
Learning objectives	1. Distinguish Object-Oriented programming paradigm from Procedure-Oriented Programming 2. Use the Java programming language for various programming technologies.

Course Outcomes:

On the successful completion of the course, the student will be able to

COs	**Course Outcomes**	**Bloom's level**
CO1	Explain the principles of Object-Oriented Programming using Java constructs.	Understanding
CO2	Develop Java programs using classes, objects, Inheritance & Interfaces, packages, exception handling, and multithreading to solve computational problems.	Apply
CO3	Analyze real-world problem scenarios to identify appropriate object-oriented features (inheritance, interfaces, polymorphism) and justify their use in program design.	Analyze
CO4	Evaluate the performance, readability, and robustness of Java programs by comparing different coding approaches and debugging strategies.	Evaluate

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12		PSO1	PSO2
CO1	3	2	2	-	-	-	-	-	-	-	-	2		2	-
CO2	3	3	3	2	2	-	-	-	2	2	-	2		2	2
CO3	3	3	2	2	2	-	-	-	2	-	-	2		3	2
CO4	3	3	3	2	2	-	-	1	1	2	-	3		3	3

Mapping Strength: Strong– 3 Medium – 2 Low – 1

Course Structure

Sl. No.	Modules	No. of Lecture Hours	No. of Tutorial Hours	No. of Practical Hours
	Module – 1 Overview of Java			
1.1	**An Overview of Java:** Object-Oriented Programming (Two Paradigms, Abstraction, The Three OOP Principles), Using Blocks of Code, Lexical Issues (Whitespace, Identifiers, Literals, Comments, Separators, The Java Keywords).	1	-	-
1.2	**Data Types, Variables, and Arrays:** The Primitive Types (Integers, Floating-Point Types, Characters, Booleans), Variables, Type Conversion and Casting, Automatic Type Promotion in Expressions, Arrays	1	-	-
1.3	**Operators:** Arithmetic Operators, Relational Operators, Boolean Logical Operators, The Assignment Operator, The ? Operator, Operator Precedence, Using Parentheses.	1	-	-
1.4	**Control Statements:** Java's Selection Statements (if, The Traditional switch), Iteration Statements (while, do-while, for, The For-Each Version of the for Loop), Jump Statements (Using break, Using continue, return).	1	-	2
	Module – 2: Classes and Objects			
2.1	**Introducing Classes:** Class Fundamentals**,** Declaring Objects**,** Assigning Object Reference Variables	1	-	2
2.2	Introducing Methods**,** Constructors**,** This Keyword**,** Garbage Collection.	1	-	2
2.3	**Methods and Classes:** Overloading Methods, Objects as Parameters, Argument Passing, Returning Objects, Access Control	1	-	-
2.4	Understanding static keyword, introducing final, Introducing super keyword.	2	-	-
	Module – 3: Inheritance & Interfaces			
3.1	**Inheritance:** Inheritance Basics, Creating a Multilevel Hierarchy.	1	-	-
3.2	When Constructors Are Executed, Method Overriding	1		
3.3	Using Abstract Classes	1	-	-

3.4	**Interfaces:** Interfaces, Default Interface Methods, Use static Methods in an Interface, Private Interface Methods .	1	-	2
Module – 4: Packages & Exceptions				
4.1	**Packages:** Packages, Packages and Member Access, Importing Packages.	1	-	2
4.2	**Exceptions:** Exception-Handling Fundamentals**,** Exception Types**,** Uncaught Exceptions	1	-	2
4.3	Using try and catch, Multiple catch Clauses, Nested try Statements,	1	-	-
4.4	Throw, throws, finally, Java's Built-in Exceptions, Creating Your Own Exception Subclasses.	1	-	-
Module – 5: Threads				
5.1	**Multithreaded Programming:** The Java Thread Model, The Main Thread, Creating a Thread, Creating Multiple Threads,	1	-	2
5.2	Using is Alive() and join(), Thread Priorities	1	-	1
5.3	Synchronization	1	-	-
Total No. of Lecture Hours		25		
Total No. of Tutorial Hours			-	-

Integrated lab program List

Sl. No.	Programs	COs
1.	For Practice only – not added for SET (Input -using Command Line Arguments) a) Develop a JAVA program to display the message. b) Develop a JAVA program to implement basic arithmetic operations. c) Develop a JAVA program to count and display the number of arguments.	CO1
2.	Develop a JAVA program to add TWO matrices of suitable order N (The value of N should be read from command line arguments).	CO1
3.	A class called Employee, which models an employee with an ID, name and salary, is designed as shown in the following class diagram. The method raiseSalary (percent) increases the salary by the given percentage. Develop the Employee class and suitable main method for demonstration.	CO2
4.	A class called MyPoint, which models a 2D point with x and y coordinates, is designed as follows: ● Two instance variables x (int) and y (int). ● A default (or "no-arg") constructor that construct a point at the default location of (0, 0). ● A overloaded constructor that constructs a point with the given x and y coordinates. ● A method setXY() to set both x and y. ● A method getXY() which returns the x and y in a 2-element int array. ● A toString() method that returns a string description of the instance in the format "(x, y)". ● A method called distance(int x, int y) that returns the distance from this point to another point at the given (x, y) coordinates ● An overloaded distance(MyPoint another) that returns the distance from this point to the givenMyPoint instance (called another) ● Another overloaded distance() method that returns the distance from this point to the origin (0,0)	CO2

	Develop the code for the class MyPoint. Also develop a JAVA program (called TestMyPoint) to test all the methods defined in the class.	
5.	Develop a JAVA program to create a class named shape. Create three sub classes namely: circle, triangle and square, each class has two member functions named draw () and erase (). Demonstrate polymorphism concepts by developing suitable methods, defining member data and main program.	CO3
6.	Develop a JAVA program to create an interface Resizable with methods resizeWidth(int width) and resizeHeight(int height) that allow an object to be resized. Create a class Rectangle that implements the Resizable interface and implements the resize methods.	CO3
7.	Develop a JAVA program to raise a custom exception (user defined exception) for DivisionByZero using try, catch, throw and finally.	CO4
8.	Develop a JAVA program to create a package named mypack and import & implement it in a suitable class.	CO4
9.	Write a program to illustrate creation of threads using runnable class. (start method start each of the newly created thread. Inside the run method there is sleep() for suspend the thread for 500milliseconds).	CO5
10.	Develop a program to create a class MyThread in this class a constructor, call the base class constructor, using super and start the thread. The run method of the class starts after this. It can be observed that both main thread and created child thread are executed concurrently.	CO5

Text book:

1. Herbert Schildt, ***Java The Complete Reference-Eleventh Edition***, McGraw Hill; 12th Edition.

Reference Book:

1. Dr. R. Nageswara Rao, Core Java, An Integrated Approach, Dream tech Press, 2016.
2. Mahesh Bhave and Sunil Patekar, "Programming with Java", First Edition, Pearson Education, 2008
3. Rajkumar Buyya, S Thamarai selvi, xing chenchu, Object oriented Programming with java, Tata McGraw Hill Education Private Limited, 2009
4. Richard A Johnson, An Introduction to Java Programming and Object-Oriented Application Development, Delmar Cengage Learning, 2007.
5. Michael Blaha, James Rumbaugh: Object-Oriented Modeling and Design with UML, 2nd Edition, Pearson Education/ PHI, 2007.
6. E Balagurusamy , Programming with Java, 6th Edition, by, Mar-2019, McGraw Hill Education, ISBN: 9789353162337.
7. Bruce Eckel Thinking in Java, Fourth Edition, Prentice Hall, 2006(https://sd.blackball.lv/library/thinking_in_java_4th_edition.pdf)

Online Resources:

1. Java Tutorial: https://www.geeksforgeeks.org/java/
2. Introduction To Programming In Java (by Evan Jones, Adam Marcus and Eugene Wu):
3. https://ocw.mit.edu/courses/6-092-introduction-to-programming-in-java-january-iap-2010/
4. Java Tutorial: https://www.w3schools.com/java/
5. Java Tutorial: https://www.javatpoint.com/java-tutorial
6. https://www.udacity.com/course/java-programming-basics--ud282

Course Code: BCS306B **Course: Object Oriented Programming with C++**

Credits: 3 **CIE: 50 Marks**

L:T:P - 2:0:2 **SEE: 50 Marks**

SEE Hours: 3 **Total Marks: 100**

Prerequisites if any	**C Programming**
Learning objectives	1. To provide the knowledge of fundamental principles of Object-Oriented Programming. 2. To introduce the concepts of C++ language such as class, object, inheritance, overloading, virtual functions, STL etc.

Course Outcomes:

On the successful completion of the course, the student will be able to

COs	**Course Outcomes**	**Bloom's level**
CO1	Explain principles of Object-Oriented Programming using class and objects.	Understanding
CO2	Illustrate the concepts of function and operator overloading.	Apply
CO3	Demonstrate reusability using inheritance and virtual functions.	Apply
CO4	Examine real-world problems to determine appropriate solutions and utilize function templates, class templates, STL, and exception handling for creating efficient and reliable C++ applications.	Analyze

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12		PSO1	PSO2
CO1	3	2	2	-	2	-	-	-	2	2	-	2		-	-
CO2	3	3	2	-	2	-	-	-	2	2	-	2		3	2
CO3	3	3	2	-	2	-	-	-	2	2	-	2		3	2
CO4	3	3	2	2	2	-	-	-	3	3	-	3		3	3

Mapping strength: 3 – Strong 2 – Medium 1 – Low

Course Structure

No s	**Modules**	**No of Lecture Hours**	**No of Tutorial Hours**	**No of Practical Hours**
	Module-1: Classes and Objects			
1.1	**Classes and Objects:** An overview of C++,classes and objects, Relationship of Structure, Union and Class in C++	1	-	-
1.2	Friend Functions, Friend Classes, Inline Functions- Defining Inline Functions Within a Class	1	-	-
1.3	Constructors and Destructors, Parameterized Constructors-Constructors with One Parameter: A Special Case, copy constructor.	1	-	-

1.4	Static Class Members- Static Data Members and Static Member Functions, When Constructors and Destructors Are Executed, Passing Objects to Functions, Returning Objects.	1	-	1
1.5	**Arrays, pointers, References, and the Dynamic Allocation Operators:** _this‘ Pointer, References – Reference Parameters, Passing References to Objects and Returning References.,	1	-	1
1.6	C++_s Dynamic Allocation Operators- Initializing Allocated Memory, Allocating Arrays and Allocating Objects	1	-	1
	Module-2: Overloading			
2.1	**Function Overloading, Copy Constructor, and Default Arguments:** Function Overloading	1	-	-
2.2	Copy Constructor, Default Function Arguments, Default Argument vs. Overloading.	1	-	1
2.3	**Operator Overloading:** Creating a Member Operator Function- Creating Prefix and Postfix Forms of the Increment and Decrement Operators	1	-	1
2.4	Operator Overloading Using a Friend Function – Using a Friend to Overload ++ or --, Friend operator Functions Add Flexibility, Overloading << and >>.	2	-	1
	Module – 3: Inheritance			
3.1	**Inheritance:** Base-Class Access Control, Inheritance and protected Members- Protected Base-Class Inheritance, Inheritance Multiple Base Classes	2	-	1
3.2	Constructors, Destructors, and Inheritance- When Constructors and Destructors Are Executed, Passing Parameters to Base-Class Constructors. Granting Access, Virtual Base Classes.	1	-	1
3.3	**Virtual Functions and Polymorphism:** Virtual Functions- Calling a Virtual Function Through a Base- Class Reference, The Virtual Attribute vs. Inherited,	1	-	1
3.4	Virtual Functions Are Hierarchical, Pure Virtual Functions- Abstract Classes, Using Virtual Functions, Early vs. Late Binding	1	-	1
	Module – 4: Generic functions			
4.1	**Templates:** Generic Functions- A Function with Two Generic Types, Explicitly Overloading a Generic Function,	1	-	-
4.2	Overloading a Function Template, Using Standard Parameters with Template Functions, Generic Function Restrictions.	1	-	1
4.3	**Applying Generic Function:** A generic Sort.	2	-	-
4.4	**Applying Generic Classes:** An Example with Two Generic Data Types, A Generic Array Class, Using Non-Type Arguments with Generic classes, Using Default Arguments with Template Classes.	1	-	1
	Module – 5: Exception handling			
5.1	**Exception Handling:** Fundamentals, Handling derived class Exception	1	-	1
5.2	Exception Handling options.	1	-	1

5.3	**Introducing the Standard Template Library:** An Overview of the STL, Container Classes,	1	-	-
5.4	General Theory of Operation, Vector container class, Algorithms.	1	-	1
	Total No. of Lecture Hours	**25**	-	**-**
	Total No. of Tutorial Hours		**-**	**-**
	Total No. of Practical Hours			**15**

Integrated Lab Programs		
Sl. No.	**Programs**	**COs**
1.	a. Write a C++ program to read three numbers from the keyboard and display the largest value on the screen. b. Write a C++ program to check whether the given number is an Armstrong number or not.(Hint: Armstrong number 153=1^3 + 5^3 + 3^3). c. Write a C++ program to find and display Factorial of a number. (Using an iterative method or recursion). d. Write a C++ program to print the ASCII value of a character. Write a C++ program that will accept an array of numbers and display the number of times the given number occurred in the array.	CO1
2.	a) Convert time from HH:MM:SS format to seconds using class in C++. b) A phone number, such as (044) 234-8900, can be thought of as having three parts: the area code (044), the exchange (234) and the number (8900). Write a program that uses a class to store these three parts of a phone number separately. Call the class phone. Create two class objects of type phone. Initialize one, and have the user input a number for the other one. Display both the numbers.	CO1
3.	Create two classes DM and DB which store the value of distances. DM stores distance in meters and centimeters and DB in feet and inches. Write a program that can read values for the class objects and add one object of DM with another object of DB. Use a friend function to carry out the addition operation. The object that stores the results may be a DM object or DB object, depending on the units in which the results are required. The display should be in the format of feet and inches or meters and centimeters depending on the object on display.	CO1
4.	Write a C++ program to create a class Rectangle with data members: length, breadth, area and member functions as: Rectangle&setDimensions(const int&, const int&) – sets the length and breadth of arguments using constant references. Use this pointer to return the resized Rectangle by reference. i. int computeArea() – to compute and returns area of a rectangle. ii. int computePerimeter() – to compute and return perimeter of a rectangle. iii. Two constructors, default constructor to initialize data members to zero and an overloaded constructor as arguments with breadth having a default value. Define all the member functions outside the class. Create objects of Rectangle type and test all the functions. Delete the objects before the program terminates.	CO1
5.	Write a C++ program to create a class called Complex and implement the following overloading member functions that return a Complex number after performing addition of input arguments. ● Complex ADD (int a, Complex s) – where a is an integer (real part) and s is a complex number. ● Complex ADD (Complex &s1, Complex &s2) Create objects of Rectangle type on heap utilizing both constructors. Use the member functions to modify the rectangle size, compute area, perimeter. Display the area & perimeter of each rectangle. Delete the objects before program termination.	CO2

6.	Write a C++ program to create a class called STACK using an array of integers. Implement the following operations by overloading the operators _+‘ and _-‘. S1 = S1 + element; where S1 is the object of class STACK and element is an integer to be pushed on the top of stack. int element = S1--; where S1 is the object of class STACK. _—_operator pops the top element. Handle the STACK empty and full conditions and also display the contents after every operation by overloading << operator.	CO2
7.	Write a C++ program to read and print Employee information (name, empID, gender) with Department (deptName, workAssigned) and with Loan information (loanDetails,loanAmt) using hierarchical inheritance.	CO3
8.	Write a C++ program to design a Student class representing USN and a Test class representing the scores of the student in various subjects and a Sports class representing the score in sports. The Sports and Test classes is inherited by Result class having the functionality to add the scores and display the final result of a student.	CO3
9.	Write a C++ program to create a class called STUDENT with data members USN, Name and Age. Using inheritance, create the classes UGSTUDENT and PGSTUDENT having fields as Semester, Fees and Stipend. Enter the data for at least 5 students from UG and PG. Find the average age for all UG and PG students separately.	CO3

10.	Implement class Shape with the following specification:
	class Shape{
	protected:
	float area, perimeter; public:
	Shape();
	virtual void initialize()=0; virtual float computeArea()=0;
	virtual float computePerimeter()=0; virtual
	~Shape();
	};
	Implement 2 classes Triangle and Rectangle publicly derived from class Shape, with suitable data members. Implement all the functions derived from class shape in each of the derived classes. Write a C++program to create objects of each of the derived class and assign to the base class (Shape) type pointer/reference. Demonstrate runtime polymorphism by calling the functions of the derived class objects by using the base class pointer/reference.
		CO3

11.	Write two function templates in C++ to . Sort the numbers. a. To search a given number Demonstrate the above functions on an array of integers and double. b. A point on the 2D can be represented by two numbers: an x co-ordinate and a y co- ordinate. The sum of two points can be defined as a new point whose x co-ordinate is the sum of x co-ordinates of both points and same for y co-ordinates. Using function template, find the third point in C++.	CO4
12.	a) Write a simple calculator using class template in C++ b) Write a program implementing stack and its operations using template class.	CO4

13.	Write a C++ program with the following: . A function to read two double type numbers from the keyboard. a. A function to calculate the division of these two numbers. b. A try block to throw an exception when a wrong type of data is keyed in. c. A try block to detect and throw an exception if the condition —divide-by-zero occurs. Appropriate catch block to handle the exceptions thrown.	CO4
14.	a. Perform these basic vector operation using Standard Template Library: . Find the number of elements in the vector. i. Check whether the vector is empty or not. ii. Insert some elements into the vector. iii. Remove the element at a particular position. iv. Find the index of a particular element in a vector. b. Make a vector of random numbers and sort it in descending order using STL and also find its sum.	CO4

Text Books:

1. C++ The Complete Reference, Herbert Schildt, TMH, McGraw-Hill, 4th Edition, 2017.

Reference Books:

2. The C++ programming language, Bjarnestroustrup , Pearson Education, 3rd Edition, 2013.
3. C++ Primer, Stanley B.Lippman and Josee Lajore, Addison Wesley, 3rd Edition, 2014.
4. WEBLINK: http://www.cplusplus.com/

Online Resources:

1. NPTEL: Programming in modern C++ https://nptel.ac.in/courses/noc22_cs43

   Coursera: Programming in C++- https://www.coursera.org/specializations/hands-on-cpp

Code: BCS306D **Course: JavaScript**

Credits: 3 **CIE: 50 Marks**

L:T:P - 2:0:2 **SEE: 50 Marks**

SEE Hours: 3 **Total Marks:100**

Prerequisites if any	Programming Fundamentals
Learning objectives	Learners will be able to: ● Compose syntactically correct JavaScript programs, including a functional "Hello World" example. They will confidently apply essential lexical elements such as comments, literals, and identifiers. ● Proficiently manipulate JavaScript's core data structures. They will aptly create, modify, and query object properties, employ functions to solve tasks, comprehend closures for encapsulation, and effectively utilize the JavaScript Standard Library to solve challenges involving sets, maps, regular expressions, and error handling.

Course Outcomes:

On the successful completion of the course, the student will be able to

COs	**Course Outcomes**	**Bloom's level**
CO1	Apply the knowledge of JavaScript lexical elements, data types, values, and variables to write programs.	Apply
CO2	Design and develop interactive web applications using object-oriented concepts, various control structures for statements and loops, and effectively manage data using arrays and objects.	Analyze
CO3	Create programs which uses Document Object Model, manipulate web documents and styles dynamically to enhance user experiences.	Apply

Mapping with POs and PSOs:

COs	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**		**PSO1**	**PSO2**
CO1	3	2	-	-	3	-	-	-	-	-	-	2		3	3
CO2	3	2	-	-	3	-	-	-	-	-	-	2		3	3
CO3	3	2	-	-	3	-	-	-	-	-	-	3		3	3

Mapping Strength: Strong– 3 Medium – 2 Low – 1

Course Structure

Sl. no	**Modules**	**No. of Lecture Hours**	**No. of Tutorial Hours**	**No. of Practical Hours**
Module – 1: Introduction to JavaScript				
1.1	Introduction to JavaScript: Exploring JavaScript, Hello World, A Tour of JavaScript	1	-	1
1.2	Lexical Structure: The Text of a JavaScript Program, Comments, Literals, Identifiers and Reserved Words, Unicode, Optional Semicolons	2	-	1

1.3	Types, Values, and Variables: Numbers, Text, Boolean Values, null and undefined, Symbols, The Global Object, Immutable Primitive Values and MutableObject References, Type Conversions	2	-	1
Module – 2: Expressions, Operators and Statements				
2.1	Expressions and Operators: Primary Expressions, Object and Array Initializers, Function Definition Expressions, Property Access Expressions, Invocation Expressions, Arithmetic Expressions, Relational Expressions, Logical Expressions, Assignment Expressions, Evaluation Expressions,	3	-	2
2.2	Statements: Expression Statements, Compound and Empty Statements, Conditionals, Loops, Jumps, Declarations	2	-	1

Module – 3: Objects, Arrays and functions				
3.1	Objects: Introduction to Objects, Creating Objects, Querying and Setting Properties, Deleting Properties, Testing Properties, Enumerating Properties, Extending Objects, Serializing Objects, Object Methods	2	-	1
3.2	Arrays: Creating Arrays, Reading and Writing Array Elements, Sparse Arrays, Array Length, Adding and Deleting Array Elements, Iterating Arrays, Multidimensional Arrays, Array Methods, Array-Like Objects, Strings as Arrays	2	-	1
3.3	Functions: Defining Functions, Invoking Functions, Function Arguments and Parameters, Functions as Values, Functions as Namespaces, Closures, Function Properties, Methods and Constructor	2	-	1
Module – 4: Classes				
4.1	Classes: Classes and Prototypes, Classes and Constructors, Classes with the class Keyword, Adding Methods to Existing Classes, Subclasses	2	-	2
4.2	The JavaScript Standard Library: Sets and Maps, Typed Arrays and Binary Data, Pattern Matching with Regular Expressions, Dates and Times, Error Classes	3	-	1
Module – 5: Iterators and Generators				
5.1	Iterators and Generators: How Iterators Work, Implementing Iterable Objects, Generators	2	-	1
5.2	JavaScript in Web Browsers: Web Programming Basics, Events, Scripting Documents, Scripting CSS, Document Geometry and Scrolling, Web Components	2	-	2
	Total No. of Lecture Hours	**25**	-	-
	Total No. of Tutorial Hours		**0**	-
	Total No. of Practical Hours			**15**

SL.NO	COs	List of Experiments
1.	CO1	**Type Conversions** a. Create a program that demonstrates different types of type conversions in JavaScript. b. Convert a number to a string, a string to a number, and a boolean to a number.
2.	CO2	**Object Properties** Create an object representing a person with properties like name, age, and occupation. Access and modify these properties using both dot notation and bracket notation.
3.	CO2	**Array Manipulation** Write a program that demonstrates various array operations. Create an array of numbers, add new elements to it, remove elements, and iterate through the array using loops and array methods.
4.	CO2	**Function and Closure** Define a function that takes two parameters and returns their sum. Create a closure that uses a variable from its containing function's scope.
5.	CO2	**Classes and Inheritance** Create a class representing a basic shape. Extend the class to create subclasses like Circle and Square. Add methods to calculate their areas.
6.	CO2	**Error Handling** Write a program that uses a try-catch block to handle an error. Attempt to access a property of an undefined object and handle the resulting exception.
7.	CO1	**String Manipulation** Write a program that takes a sentence as input and reverses the order of words using string manipulation techniques.
8.	CO1	**Working with Dates** Create a program that displays the current date and time. Format it in a user-friendly way, such as "August 10, 2023, 3:30 PM".
9.	CO3	**Event Handling and DOM Manipulation** Create a simple to-do list. Allow users to add items, mark items as completed, and remove items using event listeners and DOM manipulation.
10.	CO2	**Generators** Implement a simple generator that yields a sequence of even numbers up to a specified limit.
11.	CO2	**Regular Expressions for Data Validation** Create a form with input fields for email, password, and phone number. Use regular expressions to validate the user's input in real-time as they type.
12.	CO3	**Creating a Basic Web Component** Explore creating a basic web component using the `CustomElementRegistry`. Build a custom element for a simple counter that users can interact with.

13.	CO1, CO2, CO3	**Project-based experiments (Students can choose any one of the following) (CO1, CO2 and CO3)** **a. Task Manager Web App** - Build a task manager web application using HTML, CSS, and JavaScript. The app should have the following features: ● Task List: Display a list of tasks with their titles and due dates. ● Add Task: Allow users to add new tasks with titles and due dates. ● Mark Complete: Users can mark tasks as completed, which will visually differentiate them from incomplete tasks. ● Delete Task: Implement a way to delete tasks from the list. ● Local Storage: Store tasks in the browser's local storage so that they persist even after the user refreshes the page. This project will require you to work with HTML for creating the structure, CSS for styling, and JavaScript for handling user interactions, managing tasks, and storing data. OR **b. Interactive Quiz Game -** Create an interactive quiz game using HTML, CSS, and JavaScript. The app should have the following features: ● Quiz Questions: Prepare a set of multiple-choice questions along with their correct answers. ● Display Questions: Display one question at a time with its options. ● Answer Selection: Allow users to select an answer from the provided options. ● Score Tracking: Keep track of the user's score based on the number of correct answers. ● Results: After all questions are answered, display the user's total score and a congratulatory message. To make the quiz more engaging, you could also implement a countdown timer for each question. Additionally, consider using animations to transition between questions.

Textbook:

1. JavaScript: The Definitive Guide, David Flanagan, 7th Edition (Released May 2020), O'Reilly Media,

Reference Book:

1. JavaScript from Beginner to Professional, Laurence Lars Svekis, Maaike van Putten, Rob Percival, Packt Publishing Limited, 2021.

Online Resources:

1. https://ocw.mit.edu/courses/6-170-software-studio-spring-2013/pages/lecture-notes/
2. https://web.stanford.edu/class/cs98si/courseinfo.html

Course Code: BSCK307 **Course: Social Connect and Responsibility**

Credits: 1 **CIE: 100 Marks**

L:T:P - 0:0:2 **Total Marks: 100**

Prerequisites if any	NIL
Learning objectives	• To make students understand and appreciate the important societal/ environmental issues like conservation/ sustainability/ waste management and the like and inculcate ethical responsibility towards the same. • Provide a formal platform for students to communicate and connect to their surroundings and enable them to have a responsible connection with society.

Course Outcomes:

On the successful completion of the course, the student will be able to

COs	Course Outcomes	Bloom's level
CO1	Interpret social responsibility	Understand
CO2	Apply sustainability and creativity	Apply
CO3	Demonstrate planning and organizational skills	Apply

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12		PSO1	PSO2
CO1	-	-	-	-	-		2	1	3	1	1	-		1	1
CO2	-	-	-	-	-		2	1	3	2	2	-		1	1
CO3	-	-	-	-	-		2	1	3	3	3	-		2	1

S – Strong (3) M – Medium (2) L – Low (1)

Course Structure

Nos.	Module Name	No. of Lecture Hours	No. of Tutorial Hours	No. of Practical Hours
Module – 1: Plantation /Heritage Walk				
1.1	Plantation and adoption of a tree: Plantation of a tree that will be adopted for four years by a group of B.E. students. They will also make an excerpt either as a documentary or a photo blog describing the plant's origin, its usage in daily life, and its appearance in folklore and literature (**After Plantation of specific tree like sandalwood or any medicinal plants mention the significance of the plants one to three pl**ants**)**	01	0	04
OR				
1.1	Heritage walk and crafts corner: Heritage tour, **knowing the history and culture of the city,** connecting to people around through their history, knowing the city and its craftsmen, photo blog and documentary on evolution and practice of various craft forms.	01	0	04
Module – 2: Organic farming/Food walk				

2.1	Organic farming and waste management: usefulness of organic farming, wet **waste management in** neighboring villages, and implementation **in the campus**.	01	0	04
	OR			
2.2	Food Walk; City's culinary practices, food lore, and indigenous materials of the region used in cooking.	01	0	04
	Module – 3: Water Conservation			
3.1	**Water Conservation**: knowing the present practices in the surrounding villages and implementation in the campus, **documentary or photo blog presenting the current practices**	01	0	04
Total No. of Hours		**15**		

Activities:

Jamming session, open mic, and poetry: Platform to connect to others. Share the stories with others. Share the experience of Social Connect: Exhibit the talent like playing instruments, singing, one-act play, art-painting, and fine art.

Pedagogy:

The pedagogy will include interactive lectures, inspiring guest talks, field visits, social immersion and a course project.

Applying and synthesizing information from these sources to define the social problem to address and try to arrive at the solution through the course project, of your group.

Social immersion with NGOs/social sections will be a key part of the course.

Guideline for Assessment Process: Continuous Internal Evaluation (CIE)

After completion of the social connect, the student shall prepare, with daily diary as reference, a comprehensive report in consultation with the mentor/s to indicate what he has observed and learnt in the social connect period. The report should be signed by the mentor. The course shall be evaluated on the basis of the following criteria and/or other relevant criteria pertaining to the activity completed / based on the rubrics approved by the DC.

Components	Marks
Marks allotted for the diary	20
Planning and scheduling the social connect	10
Information/Data collected during the social connect	20
Analysis of the information/data and report writing	30
Final presentation	20
Total	100

ABILITY ENHANCEMENT COURSE

Code: BCS358A **Course: Data Analytics with Excel**

Credits: 1 **CIE: 50 Marks**

L:T:P - 1:0:0 **SEE: NA**

SEE Hours: NA **Total Marks:50**

Prerequisites if any	Basic Computer Skills, Fundamental Excel Skills
Learning objectives	The learning objectives for a data analytics course with Excel typically revolve around equipping students with the knowledge and skills to effectively analyze and interpret data using Excel as a tool

Course Outcomes: *On the successful completion of the course, the student will be able to*

COs	Course Outcomes	Bloom's level
CO1	Describe the Basic features of Excel	Understand
CO2	Investigate data visualization tools for data analysis using MS Excel.	Analyze

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12		PSO1	PSO2
CO1	2	-	-	-	2	-	-	-	-	-	-	2		-	-
CO2	3	3	3	2	3	-	-	-	-	-	-	2		-	-

Mapping Strength: Strong– 3 Medium – 2 Low – 1

Course Structure

Sl. No.	Module	No. of Lecture Hours	No. of Tutorial Hours	No. of Practical Hours
	Module – 1: I Basics of MS Excel			
1.1	Features of MS	2	-	-
1.2	Excel Worksheets and Workbooks: Labeling and Naming Worksheets and Workbooks, Adding, Deleting and Saving Worksheets and Workbooks, Reposition Worksheets, Inserting, Deleting, and Renaming Worksheets, Copy Worksheets, Printing a Workbook, Formatting a Worksheet, Adding Elements to a Workbook, Protecting Worksheet and Workbook.	2	-	-
	Module – 2: Data Representation using MS Excel			
2.1	Import externa data, Creating a Table, Sorting Data into a Table, Data Validation, Consolidation	1	-	-
2.2	Defining Names in MS Excel, Macros: View Macros, Record Macros	2	-	-
2.3	Formulas and Functions: Creating a Formula, Formula Auditing, Meaning and Advantages of functions, Insert function, Use relative References, Mathematical Functions, Statistical Functions, Date & Time Functions	2	-	-
	Module – 3: Data Visualization and Analysis			
3.1	Charts: Chart elements: Titles, legend, data labels, creating a New Chart, Formatting the Chat, Types of charts, Using Chart Templates	2	-	-

3.2	PivotTables: Creating a PivotTable, Filtering and Sorting a PivotTable, Using Slicers to manipulate PivotTables, Creating a PivotChart	2	-	-
3.3	Filtering Data: Creating a Custom AutoFilter, Using an Advanced Filter. Data Group, Ungroup and Subtotals	2	-	-
	Total No. of Lecture Hours	**15**	-	-
	Total No. of Tutorial Hours		**00**	**-**
	Total No. of Practical Hours			**00**

Textbook:

1. "Microsoft Excel 2019 Step by Step" by Curtis Frye
2. "Excel 2019 Bible" by Michael Alexander, Richard Kusleika, and John Walkenbach

Reference Book:

1. "Microsoft Excel Data Analysis and Business Modeling" by Wayne L. Winston:
2. "Excel 2019 All-in-One For Dummies" by Greg Harvey:

Online Resources:

1. https://www.academia.edu/42074058/Excel_2019_BIBLE
2. https://ptgmedia.pearsoncmg.com/images/9780735681019/samplepages/9780735681019.pdf

Code: BCS358B **Course: Data Analytics with R**

Credits: 1 **CIE: 50 Marks**

L:T:P - 1:0:0 **SEE: NA**

SEE Hours: NA **Total Marks:50**

Prerequisites if any	Basics of any programming language and knowledge of statistics and mathematics.
Learning objectives	1. Understand the basics of data analytics and its importance in decision-making processes. 2. Utilize the R programming language for data manipulation and analysis tasks.

Course Outcomes:

On the successful completion of the course, the student will be able to

COs	Course Outcomes	Bloom's level
CO1	Apply the R programming language to perform data analysis tasks.	Analyze
CO2	Apply statistical concepts and techniques to analyze data.	Analyze

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12		PSO1	PSO2
CO1	2	2	2	3	3	-	-	-	-	-	-	3		2	2
CO2	2	2	2	3	3	-	-	-	-	-	-	3		2	2

Mapping Strength: Strong– 3 Medium – 2 Low – 1

Course Structure

Sl. No.	Module	No. of Lecture Hours	No. of Tutorial Hours	No. of Practical Hours
Module – 1: Introduction to R				
1.1	Introduction, Downloading and Installing Data Types in R.	1	-	-
1.2	RIDEs and Text Editors, Handling Packages in R, Working with Directory.	1	-	-
1.3	Few Commands for Data Exploration.	2	-	-
Module – 2: Loading and Handling Data in R				
2.1	Introduction, Challenges of Analytical Data Processing, Expression, Variables and Functions.	2	-	-
2.2	Missing Values Treatment in R, Using the _as‘ Operator to Change the Structure of Data.	2	-	-
2.3	Vectors, Few Common Analytical Tasks, Methods for Reading Data.	1	-	-
Module-3:Exploring Data in R				
3.1	Introduction, Data Frames, R Functions for Understanding Data in Data Frames.	2	-	-
3.2	Load Data Frames, Exploring Data.	2	-	-
3.3	Descriptive Statistics, Spotting Problems in Data with Visualization.	2	-	-
	Total No. of Lecture Hours	**15**	-	-
	Total No. of Tutorial Hours		**00**	-
	Total No. of Practical Hours			**00**

Textbook:

1. Data Analytics Using R by Seema Acharya, MC GRAW HILL INDIA, © Published: April 20, 2018.

Reference Book:

1. Jared P Lander, R for everyone: advanced analytics and graphics, Pearson Education, 2013

Online Resources:

1. https://dokumen.pub/data-analytics-using-r-paperback-jan-01-2018-seema-acharya-9352605241-9789352605248.html

Code: BCS358C **Course: Project Management with Git**

Credits: 1 **CIE: 50 Marks**

L:T:P -1:0:0 **SEE: NA**

SEE Hours: NA **Total Marks:50**

Prerequisites if any	NIL
Learning objectives	1. To make the students understand the basic concepts related to Version controller with Git. 2. To provide knowledge of Installation procedure and basic commands in Git.

Course Outcomes:

On the successful completion of the course, the student will be able to

COs	Course Outcomes	Bloom's level
CO1	Describe basic concepts related to Version controller with GiT.	Understanding
CO2	Explain Installation procedure and basic commands in GiT.	Understanding

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12		PSO1	PSO2
CO1	3	2	-	-	2	-	-	-	-	-	2	2		-	-
CO2	3	2	-	-	2	-	-	-	-	-	2	2		-	-

Mapping Strength: Strong– 3 Medium – 2 Low – 1

Course Structure

Sl. No.	Module	No. of Lecture Hours	No. of Tutorial Hours	No. of Practical Hours
	Module – 1			
1.1	Setting Up and Basic Commands Initialize a new Git repository in a directory. Create a new file and add it to the staging area and commit the changes with an appropriate commit message.	1	-	-
1.2	Creating and Managing Branches Create a new branch named "feature-branch." Switch to the "master" branch. Merge the "feature-branch" into "master."	1	-	-
1.3	Creating and Managing Branches Write the commands to stash your changes, switch branches, and then apply the stashed changes.	1	-	-
1.4	Collaboration and Remote Repositories Clone a remote Git repository to your local machine.	1	-	-
	Module 2			
2.1	Collaboration and Remote Repositories Fetch the latest changes from a remote repository and rebase your local branch onto the updated remote branch.	1	-	-

1.6	Collaboration and Remote Repositories Write the command to merge "feature-branch" into "master" while providing a custom commit message for the merge	2	-	-
2.1	Git Tags and Releases Write the command to create a lightweight Git tag named "v1.0" for a commit in your local repository.	1	-	-
2.2	Advanced Git Operations Write the command to cherry-pick a range of commits from "source-branch" to the current branch.	1	-	-
Module 3				
2.3	Analysing and Changing Git History Given a commit ID, how would you use Git to view the details of that specific commit, including the author, date, and commit message?	1	-	-
2.4	Analysing and Changing Git History Write the command to list all commits made by the author "JohnDoe" between "2023-01-01" and "2023-12-31."	2	-	-
2.5	Analysing and Changing Git History Write the command to display the last five commits in the repository's history.	2		
2.6	Analysing and Changing Git History Write the command to undo the changes introduced by the commit with the ID "abc123".	1	-	-
			-	-
	Total No. of Lecture Hours	**15**	-	-
	Total No. of Tutorial Hours		**00**	**-**
	Total No. of Practical Hours			**00**

Textbook:

1. Git: Version Control for Everyone Beginner's Guide Copyright © 2013 Packt, Publishing Ravishankar Somasundaram.

Reference Book:

1. Version Control with Git by Jon Loeliger, Copyright © 2009 Jon Loeliger. All rights reserved. Printed in the United States of America. ISBN: 978-0-596-52012-0 [M]1242320486

Online Resources: https://www.udemy.com/course/introduction-to-version-control-with-git-and-github/

Code: BCS358D **Course: Data Visualization with Python**

Credits: 1 **CIE: 50 Marks**

L:T:P - 0:0:2 **SEE: NA**

SEE Hours: NA **Total Marks:50**

Prerequisites if any	Introduction to python programming
Learning objectives	To understand the basics of data visualizations techniques To understand the different plotting libraries and apply in real world scenarios.

Course Outcomes: *On the successful completion of the course, the student will be able to*

COs	Course Outcomes	Bloom's level
CO1	Describe the basics of data visualization techniques and libraries	Understand
CO2	Apply the data visualization techniques in the real world problems	Apply

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12		PSO1	PSO2
CO1	3	3	-	-	2	-	-	-	-	-	-	2		2	2
CO2	3	3	-	-	2	-	-	-	-	-	-	2		2	2

Mapping Strength: Strong– 3 Medium – 2 Low – 1

Course Structure

Sl.No	Experiments	No. of Practical Hours
List of problems for which student should develop program and execute in the Laboratory		
1	a) Write a python program to find the best of two test average marks out of three test's marks accepted from the user. b) Develop a Python program to check whether a given number is palindrome or not and also count the number of occurrences of each digit in the input number. Datatypes: https://www.youtube.com/watch?v=gCCVsvgR2KU Operators: https://www.youtube.com/watch?v=v5MR5JnKcZI Flow Control: https://www.youtube.com/watch?v=PqFKRqpHrjwFor loop: https://www.youtube.com/watch?v=0ZvaDa8eT5s While loop: https://www.youtube.com/watch?v=HZARImviDxg Exceptions: https://www.youtube.com/watch?v=6SPDvPK38tw	2 Hour
2	a) Defined as a function F as Fn = Fn-1 + Fn-2. Write a Python program which accepts a value for N (where N >0) as input and pass this value to the function. Display suitable error message if the condition for input value is not followed. b) Develop a python program to convert binary to decimal, octal to hexadecimal using functions. c) Write a Python programs for Cleaning Data, Checking for missing values and Handling the missing values. Functions:https://www.youtube.com/watchca9nw Arguments:https://www.youtube.com/watch?v=ij XMGpoMkhQ Return value: https://www.youtube.com/watch?v=nuNXiEDn44 Data_Analysis_and_Visualization_Using_Python by Dr. Ossama Embarak 2018	2 Hours
3	a) Write a Python program that accepts a sentence and find the number of words, digits, uppercase letters and lowercase letters. b) Write a Python program to find the string similarity between two given strings c) Write a Python program for Reading and Cleaning CSV data. **Sample Output:** Original string: Python Exercises Python Exercises Similarity between two said strings: **Sample Output:** Original string: Python Exercises Python Exercise Similarity between two said strings:1.0 0.967741935483871 Strings: https://www.youtube.com/watch?v=lSItwlnF0eU String functions: https://www.youtube.com/watch?v=9a3CxJyTq00 Data_Analysis_and_Visualization_Using_Python by Dr. Ossama Embarak 2018	2 Hours
4	a) Write a Python program to Demonstrate how to Draw a Bar Plot using Matplotlib. b) Write a Python program to Demonstrate how to Draw a Scatter Plot using Matplotlib.	1 Hour

	https://www.youtube.com/watch?v=RRHQ6Fs1b8w&list=PLjVLYmrlmjGcC0B_FP3bkJ-JIPkV5GuZR&index=3 https://www.youtube.com/watch?v=7ABCuhWO9II&list=PLjVLYmrlmjGcC0B_FP3bkJ-JIPkV5GuZR&index=4	
5	a) Write a Python program to Demonstrate how to Draw a Histogram Plot using Matplotlib. b) Write a Python program to Demonstrate how to Draw a Pie Chart using Matplotlib. https://www.youtube.com/watch?v=Qk7caotaQUQ&list=PLjVLYmrlmjGcC0B_FP3bkJ-JIPkV5GuZR&index=6 https://www.youtube.com/watch?v=PSji21jUNO0&list=PLjVLYmrlmjGcC0B_FP3bkJ- JIPkV5GuZR&index=7	1 Hour
6	a) Write a Python program to illustrate Linear Plotting using Matplotlib. b) Write a Python program to illustrate liner plotting with line formatting using Matplotlib. https://www.youtube.com/watch?v=UO98lJQ3QGI&list=PL-osiE80TeTvipOqomVEeZ1HRrcEvtZB_	1 Hour
7	Write a Python program which explains uses of customizing seaborn plots with Aesthetic functions. https://www.youtube.com/watch?v=6GUZXDef2U0	1 Hours
8	a) Write a Python program to explain working with bokeh line graph using Annotations and Legends. b) Write a Python program for plotting different types of plots using Bokeh. https://www.youtube.com/watch?v=HDvxYoRadcA	2 Hours
9	Write a Python program to draw 3D Plots using Plotly Libraries. https://www.youtube.com/watch?v=cCck7hCanpw&list=PLE50-dh6JzC4onX-qkv9H3HtPbBVA8M94&index=4	1 hour
10	a) Write a Python program to draw Time Series using Plotly Libraries. b) Write a Python program for creating Maps using Plotly Libraries. https://www.youtube.com/watch?v=xnJ2TNrGYik&list=PLE50-dh6JzC4onX-qkv9H3HtPbBVA8M94&index=5 https://www.youtube.com/watch?v=D35m2CdMhVs&list=PLE50-dh6JzC4onX-qkv9H3HtPbBVA8M94&index=6	2 Hours
	Total No. of Lecture Hours	**00**
	Total No. of Practical Hours	***15***

Textbook:

1. Data Analysis and Visualization Using Python, Dr. Ossama Embarak, 2018
2. Python for Data Analysis by Wes McKinney, 2nd Edition, October 2017, O'Reilly Media Publisher

Online resources

1. https://doi.org/10.1007/978-1-4842-4109-7
2. https://www.oreilly.com/library/view/python-for-data/9781491957653/

The National Institute of Engineering, Mysuru

Scheme of Teaching & Examination

Department: Computer Science and Engineering (AI and ML)

B.E. 2023 Admitted Batch

Semester: IV

Sl. No.	Type of Course	Course Code	Course Title	Teaching Department (TD)	Question Paper setting Board (PSB)	Teaching Hrs/Week				Examination				Credits
						L	T	P	S	Duration in Hours	CIE Marks	SEE Marks	Total Marks	
1	PCC/BSC	BCS401	Analysis & Design of Algorithms	TD: CS	PSB: CS	3	0	0	0	3	50	50	100	**3**
2	IPCC	BAI402	Artificial Intelligence	TD: CS	PSB: CS	3	0	2	0	3	50	50	100	**4**
3	IPCC	BCS403	Database Management Systems	TD: CS	PSB: CS	3	0	2	0	3	50	50	100	**4**
4	PCCL	BCSL404	Analysis & Design of Algorithms Lab	TD: CS	PSB: CS	0	0	2	0	2	50	50	100	**1**
5	ESC	BCS405x	ESC/ETC/PLC	TD: CS	PSB: CS	2	2	0	0	3	50	50	100	**3**
6	AEC/SEC	BCS456x	Ability Enhancement Course (AEC) - IV	TD: CS	PSB: CS	If the course is a Theory: 1	0	0	0	1	50	50	100	**1**
						If the course is a Laboratory: 0	0	2	0	2				
7	BSC	BBOC407	Biology for Engineers	TD: Chem	PSB: Chem	2	0	0	0	2	50	50	100	**2**
8	UHV	BUHK408	Universal Human Values and Professional Ethics	TD: CS	PSB: CS	1	0	0	0	2	50	50	100	**1**
9	MC	BNSK459	National Service Scheme (NSS)	NSS Coordinator		0	0	2	0	-	100	-	100	**0**
		BPEK459	Physical Education (PE) (Sports & Athletics)	PED										
		BYOK459	Yoga	Yoga Teacher										
		BMUK459	Music	Music Teacher										
										Total	**500**	**400**	**900**	**19**

PCC: Professional Core Course, PCCL: Professional Core Course laboratory, UHV: Universal Human Value Course, MC: Mandatory Course (Non-credit), AEC: Ability Enhancement Course, SEC: Skill Enhancement Course, L: Lecture, T: Tutorial, P: Practical: Skill Development Activity, CIE: Continuous Internal Evaluation, SEE: Semester End Evaluation. K :This letter in the course code indicates common to all the stream of engineering. ESC: Engineering Science Course, ETC: Emerging Technology Course, PLC: Programming Language Course

Engineering Science Course (ESC/ETC/PLC)			
BCS405A	Discrete Mathematical Structures	BCS405C	Optimization Technique
BAI405B	Metric Spaces	BAI405D	Algorithmic Game Theory
Ability Enhancement Course -IV			
BDS456A	Scala	BDS456C	MERN
BDS456B	MongoDB	BDS456D	Technical writing using LATEX

Code: BCS401 **Course: Analysis and Design of Algorithms**

Credits: 3 **L:T:P - 3:0:0**

SEE: 50 Marks **CIE: 50 Marks**

SEE Hours: 03 **Max. Marks: 100**

Prerequisites if any	Recurrence Relations, Data Structures
Learning objectives	• To equip students with methods to analyze algorithm performance using mathematical and asymptotic approaches. • To enable students to apply various algorithm design strategies to solve computational problems. • To develop the ability to analyze and evaluate the efficiency and applicability of algorithms in different contexts.

Course Outcomes:

On the successful completion of the course, the student will be able to

COs	Course Outcomes	Bloom's level
CO1	Explain the fundamental principles of algorithm analysis and asymptotic notations.	Analyze
CO2	Apply algorithm design techniques such as brute force, divide-and-conquer, decrease-and-conquer, transform-and-conquer, dynamic programming, greedy method, backtracking, and branch-and-bound to solve computational problems.	Apply
CO3	Analyze the efficiency of algorithms using mathematical tools and evaluate their suitability for different problem types.	Analyze
CO4	Evaluate and compare algorithmic strategies for solving real-world problems in terms of performance, trade-offs, and limitations.	Evaluate

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12		PSO1	PSO2
CO1	3	3	2	-	-	-	-	1	-	-	-	2		3	2
CO2	3	3	3	2	2	-	-	1	-	-	1	2		3	3
CO3	2	3	3	2	-	1	-	2	-	-	2	3		3	3
CO4	2	2	3	2	1	1	-	2	-	-	2	3		3	3

Mapping Strength: Strong– 3 Medium – 2 Low – 1

Course Structure

SI. No	Module	No. of Lecture Hours	No. of Tutorial Hours	No. of Practical Hours
	Module – 1			
1	**INTRODUCTION:**			
1.1	What is an Algorithm?	1	-	-
1.2	Fundamentals of Algorithmic Problem Solving.	2	-	-
	FUNDAMENTALS OF THE ANALYSIS OF ALGORITHM EFFICIENCY:			
1.3	Analysis Framework	1	-	-
1.4	Asymptotic Notations and Basic Efficiency Classes	1	-	-
.5	Mathematical Analysis of Non recursive Algorithms	1	-	-
1.6	Mathematical Analysis of Recursive Algorithms.	1	-	-
	BRUTE FORCE APPROACHES:			
1.7	Selection Sort	1	-	-
1.8	Sequential Search	1	-	-
	Module – 2			
	BRUTE FORCE APPROACHES (contd..):			
2.1	Brute Force String Matching	1	-	-
2.2	Exhaustive Search (Travelling Salesman problem)	1	-	-
2.3	Exhaustive Search (Knapsack Problem).	1	-	-
	DECREASE-AND-CONQUER:			
2.4	Topological Sorting.	1	-	-
	DIVIDE AND CONQUER:			
2.5	Merge Sort	1	-	-
2.6	Quick Sort	1	-	-
2.7	Strassen's Matrix Multiplication.	2	-	-
	Module – 3			
	TRANSFORM-AND-CONQUER:			
3.1	Balanced Search Trees	2	-	-
3.2	Heaps and Heapsort.	3	-	-
	SPACE-TIME TRADEOFFS:		-	-
3.3	Input Enhancement in String Matching: Horspool's Algorithm.	3	-	-
	Module – 4:			
	DYNAMIC PROGRAMMING:			
4.1	The Knapsack Problem and Memory Functions	2	-	-
4.2	Warshall's and Floyd's Algorithms.	2	-	-
	THE GREEDY METHOD:			
4.3	Prim's Algorithm	2	-	-
4.4	Kruskal's Algorithm	1	-	-
4.5	Dijkstra's Algorithm	1	-	-
	Module – 5			
	LIMITATIONS OF ALGORITHMIC POWER:			
5.1	Decision Trees	2	-	-
	COPING WITH LIMITATIONS OF ALGORITHMIC POWER:			
5.2	Backtracking (n-Queens problem)	2	-	-
5.3	Backtracking (Subset-sum problem)	2	-	-
5.4	Branch-and-Bound (Travelling Salesman Problem),	1	-	-

Total No. of Lecture Hours	**40**	**-**	**-**
Total No. of Tutorial Hours		**00**	**-**
Total No. of Practical Hours			**00**

Textbook

1. Introduction to the Design and Analysis of Algorithms, By Anany Levitin, 3rd Edition (Indian), 2017, Pearson.

Reference books

1. Computer Algorithms/C++, Ellis Horowitz, SatrajSahni and Rajasekaran, 2nd Edition, 2014, Universities Press.
2. Introduction to Algorithms, Thomas H. Cormen, Charles E. Leiserson, Ronal L. Rivest, Clifford Stein, 3rd Edition, PHI.
3. Design and Analysis of Algorithms, S. Sridhar, Oxford (Higher Education)

Online Resources:

- Design and Analysis of Algorithms: https://nptel.ac.in/courses/106/101/106101060/

Code: BAI402 **Course: Artificial Intelligence**

Credits: 4 **L: T:P - 3:0:2**

SEE: 50 Marks **CIE: 50 Marks**

SEE Hours: 3 **Max. Marks: 100**

Prerequisites if any	None
Learning objectives	1. Develop an understanding of the historical context of AI and its fundamental principles. 2. Acquire proficiency in applying basic AI principles to solve problems effectively. 3. Familiarize oneself with the methodologies of inference, perception, knowledge representation, and learning in AI.

Course Outcomes: *On the successful completion of the course, the student will be able to*

COs	Course Outcomes	Bloom's level
CO1	Utilize fundamental AI principles to address problems involving problem-solving, inference, knowledge representation, and learning.	Understanding
CO2	Evaluate search and inference algorithms within the context of problem-solving	Apply
CO3	Exhibit understanding of reasoning, uncertainty, and knowledge representation for tackling real- world problems.	Apply
CO4	Execute experiments to address problems utilizing AI techniques.	Analyze

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12		PSO1	PSO2
CO1	3	2	-	2	3	-	-	-	2	2	-	3		2	2
CO2	3	3	-	2	3	-	-	-	2	2	-	3		2	2
CO3	3	2	-	2	3	-	-	-	2	2	-	3		2	2
CO4	3	2	-	3	3	-	-	-	2	2	-	3		2	2

Mapping Strength: Strong– 3 Medium – 2 Low – 1

Course Structure

Sl. No	Modules	No. of Lecture Hours	No. of Tutorial Hours	No. of Practical Hours
	Module – 1: Introduction to AI			
1.1	**Introduction:** What is AI? Foundations of AI.	2	-	-
1.2	History of Artificial Intelligence	1	-	-
1.3	The State of the Art	1	-	-
1.4	**Intelligent Agents:** Agents and environment	1	-	1
1.5	Concept of Rationality	1	-	-
1.6	The nature of environments	1	-	-
1.7	The structure of agents	1	-	-
	Module – 2: Problem solving based on searching			
2.1	Problems solving Agents, Example problems	2	-	-
2.2	Searching for solutions	1	-	-
2.3	Uniformed Search strategies – Uniform cost search	1	-	-
2.4	Breadth First Search, Depth First Search	2	-	1
2.5	Depth Limited Search	1	-	-
2.6	Iterative Deepening Depth First	1	-	1
	Module – 3: Heuristic Search Strategies			
3.1	Heuristic functions	1	-	1
3.2	Greedy best first search	1	-	-
3.3	A* algorithm	1	-	1
3.4	Local Search & Optimization: Hill Climbing	1	-	-
3.5	Genetic Algorithms	2	-	-
	Module – 4: Logical Agents, First Order Logic, Inference in First Order Logic			
4.1	**Logical Agents:** Knowledge–based agents	1	-	-
4.2	The Wumpus world, Logic	1	-	1
4.3	Propositional logic	1	-	-
4.4	Reasoning patterns in Propositional Logic	1	-	1
4.5	**First Order Logic:** Representation Revisited, Syntax and Semantics of First Order logic	2	-	-
4.6	Using First Order logic.	1	-	-
4.7	**Inference in First Order Logic:** Propositional Versus First Order Inference	1	-	-
4.8	Unification, Forward Chaining	1	-	1
4.9	Backward Chaining, Resolution.	1	-	1
	Module – 5: Quantifying Uncertainty			
5.1	Acting under Uncertainty	1	-	-
5.2	Probability Notation	2	-	-
5.3	Inference using Full Joint Distributions	1	-	1
5.4	Independence	1	-	-
5.5	Baye's Rule and its use	2	-	-
5.6	Wumpus World Revisited	1	-	-
	Total No. of Lecture Hours	**40**	-	-
	Total No. of Tutorial Hours		**00**	**-**
	Total No. of Practical Hours			**10**

Integrated Lab Programs:

Sl. No.	COs	Programs
1.	CO1	Write a program to implement Vacuum Cleaner agent for two rooms
2.	CO2	Implement and Demonstrate Depth First Search Algorithm on Water Jug Problem
3.	CO2	Implement Iterative deepening search algorithm
4.	CO2	Solve 8 puzzle problem using BFS algorithm
5.	CO2	Implement A* Search algorithm
6.	CO2	Implementation of TSP using heuristic approach
7.	CO3	Implementation of the problem-solving strategies: either using Forward Chaining or Backward Chaining
8.	CO3	Create a knowledgebase consisting of first order logic statements and prove the given query using forward reasoning.
9.	CO4	Write a program to implement following: a. Program to calculate the probability of an event using its probability notation. b. Program to calculate marginal probability from a joint distribution. c. Program to represent a simple grid world environment similar to Wumpus World and perform basic actions.
10.	CO4	Implement Tic-Tac-Toe game using Python

Textbook:

1. Stuart J. Russell and Peter Norvig, Artificial Intelligence, 3rd Edition, Pearson,2015

Reference Book:

1. Elaine Rich, Kevin Knight, Artificial Intelligence, 3rd edition, Tata McGraw Hill,2013
2. George F Lugar, Artificial Intelligence Structure and strategies for complex, Pearson Education, 5th Edition, 2011

Online Resources:

1. Artificial Intelligence: https://nptel.ac.in/courses/106105077

Code: BCS403 **Course: Database Management Systems**

Credits: 4 **L:T:P - 3:0:2**

SEE: 50 Marks **CIE: 50 Marks**

SEE Hours: 3 **Max. Marks: 100**

Prerequisites if any	NIL
Learning objectives	• Understand fundamental database concepts, architectures, and data modeling techniques. • Apply relational and NoSQL database concepts for efficient data storage, retrieval, and management. • Analyze and evaluate database technologies through design, implementation, and application development.

Course Outcomes: *On the successful completion of the course, the student will be able to*

COs	Course Outcomes	Bloom's level
CO1	Explain database architectures, models, and conceptual data modeling techniques.	Understand
CO2	Implement normalized database schemas using ER modeling, relational algebra, SQL, and NoSQL queries.	Apply
CO3	Analyze transaction processing, concurrency control, and performance optimization techniques in databases.	Analyze
CO4	Design and evaluate a complete database-driven application that meets specified user requirements using appropriate database technologies.	Evaluate

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12		PSO1	PSO2
CO1	3	2	1	-	2	-	-	-	1	-	-	2		-	-
CO2	3	3	3	2	3	-	-	-	1	-	-	2		3	3
CO3	2	3	2	3	3	-	-	-	1	-	-	2		3	3
CO4	2	3	3	3	3	-	1	-	2	2	2	3		3	3

Mapping Strength: Strong– 3 Medium – 2 Low – 1

Course Structure

Sl. No.	Modules	No. of Lecture Hours	No. of Tutorial Hours	No. of Practical Hours
	Module – 1: Introduction to Databases			
1.1	**Introduction to Databases:** Introduction, Characteristics of database approach, Advantages of using the DBMS approach, History of database applications.	2	-	-
1.2	**Overview of Database Languages and Architectures:** Data Models, Schemas, and Instances. Three schema architecture and data independence. Database languages, and interfaces, The Database System environment.	3	-	-
1.3	**Conceptual Data Modelling using Entities and Relationships:** Entity types, Entity sets and structural constraints, Weak entity types, ER diagrams, Specialization and Generalization	3	-	1
	Module – 2: Relational Databases			
2.1	**Relational Model:** Relational Model Concepts, Relational Model Constraints and relational database schemas, Update operations, transactions, and dealing with constraint violations.	3	-	-
2.2	**Relational Algebra:** Unary and Binary relational operations, additional relational operations (aggregate, grouping, etc.) Examples of Queries in relational algebra.	3	-	-
2.3	**Mapping Conceptual Design into a Logical Design:** Relational Database Design using ER-to-Relational mapping	2	-	-
	Module – 3: Normalization and SQL			
3.1	**Normalization:** Database Design Theory – Introduction to Normalization using Functional and Multivalued Dependencies: Informal design guidelines for relation schema, Functional Dependencies, Normal Forms based on Primary Keys, Second and Third Normal Forms, Boyce-Codd Normal Form, Multivalued Dependency and Fourth Normal Form, Join Dependencies and Fifth Normal Form.	5	-	1
3.2	**SQL:** SQL data definition and data types, Schema change statements in SQL, specifying constraints in SQL, retrieval queries in SQL, INSERT, DELETE, and UPDATE statements in SQL, Additional features of SQL.	3	-	2
	Module – 4: SQL and Transactions			
4.1	**SQL:** Advanced Queries: More complex SQL retrieval queries, Specifying constraints as assertions and action triggers, Views in SQL.	3	-	2
4.2	**Transaction Processing:** Introduction to Transaction Processing, Transaction and System concepts, Desirable properties of Transactions, Characterizing schedules based on recoverability, Characterizing schedules based on Serializability, Transaction support in SQL.	5	-	1
	Module – 5: Concurrency control and NoSQL Databases			
5.1	**Concurrency Control in Databases:** Two-phase locking techniques for Concurrency control, Concurrency control based on Timestamp ordering, Multiversion Concurrency control techniques, Validation Concurrency control techniques, Granularity of Data items and Multiple Granularity Locking.	4	-	1
5.2	**NoSQL Databases and Big Data Storage Systems:** Introduction to NOSQL Systems, The CAP Theorem, Document-Based NOSQL Systems and MongoDB, NOSQL Key- Value Stores, Column-Based or Wide Column NOSQL Systems.	4	-	2
	Total No. of Lecture Hours	**40**	-	-
	Total No. of Tutorial Hours		**00**	**-**
	Total No. of Practical Hours			**10**

List of Experiments

Sl. No.	Experiments	COs
1.	Create a table called Employee & execute the following. Employee (EMPNO, ENAME, JOB, MANAGER_NO, SAL, COMMISSION) 1. Create a user and grant all permissions to the user. 2. Insert any three records in the employee table containing attributes. EMPNO, ENAME JOB, MANAGER_NO, SAL, COMMISSION and use rollback. Check the result. 3. Add primary key constraint and not null constraint to the employee table. 4. Insert NULL values to the employee table and verify the result.	CO3
2.	Create a table called Employee that contains attributes EMPNO, ENAME, JOB, MGR, SAL and execute the following. 1. Add a column commission with domain to the Employee table. 2. Insert any five records into the table. 3. Update the column details of job. 4. Rename the column of Employ table using alter command. 5. Delete the employee whose EMPNO is 105.	CO3
3.	Queries using aggregate functions (COUNT, AVG, MIN, MAX, SUM), Group by, Orderby. Employee (E_id, E_name, Age, Salary) 1. Create Employee table containing all Records E_id, E_name, Age, Salary. 2. Count number of employee names from employee table. 3. Find the Maximum age from the employee table. 4. Find the Minimum age from the employee table. 5. Find salaries of employees in Ascending Order. 6. Find grouped salaries of employees.	CO3
4.	Create a row level trigger for the customers table that would fire for INSERT or UPDATE or DELETE operations performed on the CUSTOMERS table. This trigger will display the salary difference between the old & new Salary. CUSTOMERS (ID, NAME, AGE, ADDRESS, SALARY)	CO4
5.	Create cursor for Employee table and extract the values from the table. Declare the variables, Open the cursor, and extract the values from the cursor. Close the cursor. Employee (E_id, E_name, Age, Salary)	CO4
6.	Install an Open-Source NoSQL Data base MongoDB & perform basic CRUD (Create, Read, Update & Delete) operations. Execute MongoDB basic Queries using CRUD operations.	CO5
7.	Project-based Experiment: The project should use all the database concepts covered in theory and laboratory sessions. Students can integrate other relevant concepts/technologies as required.	CO1, CO2, CO3, CO4.

Textbook:

1. Fundamentals of Database Systems, Ramez Elmasri and Shamkant B. Navathe, 7th Edition, 2017, Pearson.

Reference Book:

1. Database management systems, Ramakrishnan, and Gehrke, 3rd Edition, 2014, McGraw Hill.

Online Resources:

1. MIT OpenCourseWare Course Link: https://ocw.mit.edu/courses/6-830-database- systems-fall-2010/
2. IIT Kharagpur Course Link: https://cse.iitkgp.ac.in/~pabitra/course/dbms/dbms_new.html
3. NPTEL Course Link: https://onlinecourses.nptel.ac.in/noc22_cs91/preview

Analysis & Design of Algorithms Lab		Semester	4
Course Code	**BCSL404**	CIE Marks	50
Teaching Hours/Week (L:T:P)	0:0:2	SEE Marks	50
Credits	01	Exam Hours	2
Examination type (SEE)	Practical		

Course outcomes:

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

1. Develop programs to solve computational problems using suitable algorithm design strategy.
2. Compare algorithm design strategies by developing equivalent programs and observing running times for analysis (Empirical).
3. Make use of suitable integrated development tools to develop programs
4. Choose appropriate algorithm design techniques to develop solution to the computational and complex problems.
5. Demonstrate and present the development of program, its execution and running time(s) and record the results/inferences.

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12		PSO1	PSO2
CO1	2	3	3	-	-	-	-	-	1	1	-	2		3	2
CO2	2	3	3	-	-	-	-	-	1	2	-	2		3	2
CO3	3	2	3	2	-	1	-	-	1	2	2	2		3	2
CO4	3	2	3	2	-	-	-	-	1	2	2	2		3	2
CO5	2	2	3	-	-	-	-	-	-	-	-	3		3	2

Mapping Strength: Strong– 3 Medium – 2 Low – 1

Sl.No	CO's	Experiments
1	CO1 to CO5	Design and implement C/C++ Program to sort a given set of n integer elements using Selection Sort method and compute its time complexity. Run the program for varied values of n> 5000 and record the time taken to sort. Plot a graph of the time taken versus n. The elements can be read from a file or can be generated using the random number generator.
2	CO1 to CO5	Design and implement C/C++ Program to obtain the Topological ordering of vertices in a given digraph.
3	CO1 to CO5	Design and implement C/C++ Program to sort a given set of n integer elements using Merge Sort method and compute its time complexity. Run the program for varied values of n> 5000, and record the time taken to sort. Plot a graph of the time taken versus n. The elements can be read from a file or can be generated using the random number generator.

4	CO1 to CO5	Design and implement C/C++ Program to sort a given set of n integer elements using Quick Sort method and compute its time complexity. Run the program for varied values of n> 5000 and record the time taken to sort. Plot a graph of the time taken versus n. The elements can be read from a file or can be generated using the random number generator.
5	CO1 to CO5	Design and implement C/C++ Program to solve 0/1 Knapsack problem using Dynamic programming method.
6	CO1 to CO5	a. Design and implement C/C++ Program to solve All-Pairs Shortest Paths problem using Floyd'salgorithm. b. Design and implement C/C++ Program to find the transitive closure using Warshal's algorithm.
7	CO1 to CO5	Design and implement C/C++ Program to find Minimum Cost Spanning Tree of a given connected undirected graph using Prim's algorithm.
8	CO1 to CO5	Design and implement C/C++ Program to find Minimum Cost Spanning Tree of a given connected undirected graph using Kruskal's algorithm.
9	CO1 to CO5	Design and implement C/C++ Program to find shortest paths from a given vertex in a weighted connected graph to other vertices using Dijkstra's algorithm.
10	CO1 to CO5	Design and implement C/ C++ Program for N Queen's problem using Backtracking.

Suggested Learning Resources:

Textbook

1. Introduction to the Design and Analysis of Algorithms, By Anany Levitin, 3rd Edition (Indian), 2017, Pearson.
2. Virtual Labs (CSE): http://cse01-iiith.vlabs.ac.in/

Code: BCS405A **Course: Discrete Mathematical Structures**

Credits: 3 **L:T:P - 2:2:0**

SEE: 50 Marks **CIE: 50 Marks**

SEE Hours: 3 **Max. Marks: 100**

Prerequisites if any	Basics of number system, Mathematics
Learning objectives	1. To solve problems using concepts of Functions. 2. Solve problems using Relations and its properties. 3. To introduce Generating Functions and Recurrence Relations 4. To introduce concepts and properties of Graphs 5. To introduce the concepts of Trees and its properties.

Course Outcomes:

On the successful completion of the course, the student will be able to

COs	Course Outcomes	Bloom's level
CO1	Describe the foundational concepts of discrete mathematics including functions, recurrence relations, graphs, and trees, and explain their structures, properties, and basic applications.	Understand
CO2	Apply the principles of functions, recurrence relations, graph theory, and trees to model and solve computational and real-world problems.	Apply
CO3	Analyze problems using concepts of functions, recurrence relations, graph theory, and trees to identify patterns, relationships, and underlying structures for effective problem solving.	Analyze
CO4	Examine and critically evaluate the properties and applications of functions, recurrence relations, graph theory, and trees to make informed decisions in problem-solving scenarios.	Evaluate

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12		PSO1	PSO2
CO1	3	3	2	2	-	-	-	-	3	-	3	2		-	3
CO2	3	3	2	2	-	-	-	-	3	-	3	2		-	3
CO3	3	3	3	2	-	-	-	-	3	-	3	2		-	3
CO4	3	3	2	3	-	-	-	-	3	-	3	2		-	3

Mapping Strength: Strong– 3 Medium – 2 Low – 1

Course Structure

SI.No	Module	No. of Lecture Hours	No. of Tutorial Hours	No. of Practical Hours
	Module – 1: Functions			
1.1	**Functions:** Cartesian Products and Relations	2	-	-
1.2	Plain and One-to-One, Onto Functions	1	1	-
1.3	The Pigeonhole Principle	1	1	-
1.4	Function Composition and Inverse Functions	1	1	-
	Module – 2: Relations			
2.1	**Relations:** Properties of Relations,	1	1	-
2.2	Computer Recognition – Zero-One Matrices and Directed Graphs	2	1	-
2.3	Partial Orders – Hasse Diagrams.	2	1	-
	Module – 3: Recurrence Relations			
3.1	**Recurrence Relations:** First order linear recurrence relations,	2	-	-
3.2	The Second order linear homogeneous recurrence relation with constant coefficients	1	1	-
3.3	Non Homogeneous recurrence relation	2	1	-
	Module – 4: Graph Theory and Applications			
4.1	**Graph Theory and Applications**: Definitions and Examples Sub graphs, Complements	1	1	-
4.2	Graph Isomorphism, Vertex Degree, Euler Trails and Circuits	1	-	-
4.3	Planar Graphs	1	1	-
4.4	Hamilton Paths and Cycles	1	1	-
4.5	Graph Coloring, and Chromatic Polynomials	1	1	-
	Module – 5: Trees			
5.1	**Trees:** Definitions, Properties, and Examples	2	1	-
5.2	Rooted Trees	1	1	-
5.3	Trees and Sorting	1	1	-
5.4	Weighted Trees and Prefix Codes	1	-	-
	Total No. of Lecture Hours	**25**	-	-
	Total No. of Tutorial Hours		**15**	-
	Total No. of Practical Hours			**00**

Textbook:

1. Discrete and Combinatorial Mathematics, Ralph P. Grimaldi, 5th Edition, PHI/Pearson Education, 2004.

Reference Book:

1. Handbook of discrete and combinatorial mathematics, Kenneth H.Rosen, John G.Michels.

2. Mathematics of Computer Science, Prof. Albert R.Meyer, MIT Open Course Ware.
3. Concrete Mathematics: A foundation for computer science, Ronald L.Graham, Donald Ervin Knuth, Oren Patashnik
4. Graph Theory with Applications to Engineering and Computer Science by NarsinghDeo, Prentice-Hall, 2004

Code: BAI405B **Course: METRIC SPACES**

Credits: 03 **L:T:P - 2:2:0**

SEE: 50 Marks **CIE: 50 Marks**

SEE Hours: 03 **Max. Marks: 100**

Prerequisites if any	Recurrence Relations, Data Structures
Learning objectives	• Provide insight into the theory of sets • Learn basic concepts of metric spaces • Understand the concepts of connected sets and compact spaces

Course Outcomes:

On the successful completion of the course, the student will be able to

COs	Course Outcomes	Bloom's level
CO1	Explain basic facts about the cardinality of a set and various set-theoretic paradoxes.	Analyze
CO2	Apply the concepts of open and closed spheres and bounded sets to solve problems.	Apply
CO3	Demonstrate standard concepts of metric spaces and their properties.	Apply
CO4	Identify the continuity of a function defined on metric spaces and homomorphism.	Apply

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12		PSO1	PSO2
CO1	2	-	2	-	-	-	-	-	-	-	-	-		-	-
CO2	2	2	3	-	-	-	-	-	-	-	-	-		-	2
CO3	3	2	3	-	-	-	-	-	-	-	-	3		3	3
CO4	3	-	3	-	-	-	-	-	-	-	-	3		3	-

Mapping Strength: Strong– 3 Medium – 2 Low – 1

Course Structure

Sl no	Modules	No. of Lecture Hours	No. of Tutorial Hours	No. of Practical Hours
	Module – 1 Theory of Sets			
1.1	Finite and infinite sets, countable and uncountable sets, cardinality of sets	1	-	-
1.2	Schroder-Bernstein theorem, cantor's theorem, Order relation in cardinal numbers	1	1	-
1.3	Arithmetic of cardinal numbers, Partially ordered set	1	1	-
1.4	Zorn's lemma and axioms of choice	1	-	-
1.5	Various set-theoretic paradoxes.	1	-	-
	Module – 2 Concepts in Metric Spaces			
2.1	Definition and examples of metric spaces	1	1	-
2.2	Open spheres and Closed spheres	1	1	-
2.3	Neighborhoods, Open sets, Interior, Exterior and boundary points, Closed sets	1	1	-
2.4	Limit points and isolated points, Interior and closure of a set	1	1	-
2.5	Boundary of a set, Bounded sets, Distance between two sets, Diameter of a set.	1	1	-
	Module – 3 Complete Metric Spaces and Continuous Functions			
3.1	Cauchy and Convergent sequences, Completeness of metric spaces	1	1	-
3.2	Cantor's intersection theorem, Dense sets and separable spaces	1	1	-
3.3	Nowhere dense sets and Baire's category theorem, continuous and uniformly continuous functions	1	1	-
3.4	Homeomorphism. Banach contraction principle.	2	-	-
	Module – 4: Compactness			
4.1	Compact spaces, Sequential compactness	2	1	-
4.2	Bolzano-Weierstrass property, Compactness and finite intersection property	1	1	-
4.3	Heine-Borel theorem, Totally bounded set	1	1	-
4.4	equivalence of compactness and sequential compactness.	1	-	-
	Module – 5 Connectedness			
5.1	Separated sets, Disconnected and connected sets	1	-	-
5.2	components, connected subsets of R	2	-	-
5.3	Continuous functions on connected sets	1	1	-
5.4	Local connectedness and arc-wise connectedness	1	1	-
	Total No. of Lecture Hours	**25**	-	-
	Total No. of Tutorial Hours		**15**	-
	Total No. of Practical Hours			**0**

Text Books

1. P.K. Jain & Khalil Ahamad, "*Metric Spaces"*. Narosa, 2019.
2. Micheal O; Searcoid, "Metric spaces". Springer-Verlag, 2009.

Reference Books:

1. Satish Shirali & Harikishan L. Vasudeva, "*Metric Spaces"*, Springer-Verlag, 2006.
2. E.T. Copson, "*Metric spaces"*, Cambridge University Press, 1988.
3. P.R. Halmos, "*Naive Set Theor*y". Springer, 1974.
4. S. Kumaresan, "*Topology of Metric spaces",* 2nd edition, Narosa, 2011.
5. G.F. Simmons, "Introduction to Topology and Modern Analysis". McGraw-Hill, 2004

Online Resources:

- http://nptel.ac.in/courses.php?disciplineID=111
- http://www.class-central.com/subject/math(MOOCs)

Code: BCS405C **Course: Optimization Technique**

Credits: 3 **L:T:P - 2:2:0**

SEE: 50 Marks **CIE: 50 Marks**

SEE Hours: 03 **Max. Marks: 100**

Prerequisites if any	Recurrence Relations, Data Structures
Learning objectives	• Appreciate the importance of linear algebra in computer science and allied engineering science. • Gain the knowledge of linear algebra tools and concepts to implememt them in their core domain. ● Improve their mathematical thinking and acquire skills required for sustained lifelong learning.

Course Outcomes:

On the successful completion of the course, the student will be able to

COs	Course Outcomes	Bloom's level
CO1	Explain the concepts of vector spaces, subspaces, bases, dimension and their properties and Use matrices and linear transformations to solve the given problem	Analyze
CO2	Compute Eigenvalues and Eigenvectors for the linear transformations	Apply
CO3	Determine orthogonality of inner product spaces	Apply
CO4	Apply the optimization techniques to solve the problems.	Apply

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12		PSO1	PSO2
CO1	2	-	2	-	-	-	-	-	-	-	-	-		-	-
CO2	2	3	3	-	-	-	-	-	-	-	-	-		-	3
CO3	2	3	2	-	-	-	-	-	-	-	-	3		2	3
CO4	3	-	3	-	-	-	-	-	-	-	-	3		3	-

Mapping Strength: Strong– 3 Medium – 2 Low – 1

Course Structure

SI. No	Modules	No. of Lecture Hours	No. of Tutorial Hours	No. of Practical Hours
	Module – 1 VECTOR CALCULUS			
1.1	Functions of several variables	1	-	-
1.2	Differentiation and partial differentials	1	1	-
1.3	Gradients of vector-valued functions, gradients of matrices,	1	1	-
1.4	Useful identities for computing gradients, linearization an multivariate Taylor series.	2	1	-
	Module – 2 APPLICATIONS OF VECTOR CALCULUS			
2.1	Backpropagation and automatic differentiation	1	1	-
2.2	Gradients in a deep network	1	1	-
2.3	The Gradient of Quadratic Cost, Descending the Gradient of Cost	1	1	-
2.4	The Gradient of Mean Squared Error.	1	1	-
	Module – 3 Convex Optimization-1			
3.1	Local and global optima	1	-	-
3.2	Convex sets and functions separating	1	1	-
3.3	Hyperplanes application of Hessian matrix in optimization,	1	1	-
3.4	Optimization using gradient descent	1	1	-
3.5	Sequential search 3- point search and Fibonacci search.	1	-	
	Module – 4: Convex Optimization-2 -			
4.1	Unconstrained optimization -Method of steepest ascent/descent	2	1	-
4.2	NR method, Gradient descent	2	1	-
4.3	Mini batch gradient descent	1	1	
4.4	Stochastic gradient descent.	1	1	-
	Module – 5 Advanced Optimization			
5.1	Momentum-based gradient descent methods: Adagrad,	2	-	-
5.2	RMSprop and Adam.	2	-	-
5.3	Non-Convex Optimization: Convergence to Critical Points, Saddle-Point methods.	1	1	-
	Total No. of Lecture Hours	**25**	-	-
	Total No. of Tutorial Hours		**15**	-
	Total No. of Practical Hours			**00**

Textbook

1. Mathematics for Machine learning, Marc Peter Deisennroth, A. Aldo Faisal, Cheng Soon Ong, 2020, Cambridge University Press.
2. S. Bubeck, Convex Optimization: Algorithms and Complexity, Foundations and Trends in Optimization, 2015.
3. S. Boyd, N. Parikh, and E. Chu,“ Distributed optimization and statistical learning via the alternating direction method of multipliers”, Foundations and Trends in Machine Learning, Now Publishers Inc.

Reference books

1. Linear Algebra and Optimization for Machine Learning, Charu C. Aggarwal, Springer, 2020.
2. A.Beck, First-Order Methods in Optimization, MOS-SIAM Series on Optimization, 2017.
3. F. Bach, “Learning with Submodular Functions: A Convex Optimization Perspective”, Foundations and Trends in Machine Learning, Now Publishers Inc.

Online Resources:

- https://ocw.mit.edu/courses/mathematics/18-06sc-linear-algebra-fall- 2011/index.htm
- https://www.math.ucdavis.edu/~linear/linear.pdf
- https://www.coursera.org/learn/linear-algebra-machine-learning

Code: BAI405D **Course: ALGORITHMIC GAME THEORY**

Credits: 3 **L:T:P - 2:2:0**

SEE: 50 Marks **CIE: 50 Marks**

SEE Hours: 03 **Max. Marks: 100**

Prerequisites if any	Recurrence Relations, Data Structures
Learning objectives	• Comprehend the basics of strategic gaming and mixed strategic equilibrium. • Enable students to develop skills on extensive gaming strategies. • Analyze and discuss various gaming models. • Illustrate some real-time situations.

Course Outcomes: *On the successful completion of the course, the student will be able to*

COs	Course Outcomes	Bloom's level
CO1	Interpret the basics of strategic gaming and extensive games.	Analyze
CO2	Analyze gaming strategies on real-time incidence.	Apply
CO3	Develop the models of gaming on real-time incidence.	Apply
CO4	Apply game theory in the real world problems.	Apply

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12		PSO1	PSO2
CO1	3	-	2	-	-	-	-	-	-	-	-	-		2	-
CO2	2	3	3	-	-	-	-	-	-	-	-	-		-	3
CO3	2	3	3	-	-	-	-	-	-	-	-	2		3	3
CO4	2	-	3	-	-	-	-	-	-	-	-	3		2	-

Mapping Strength: Strong– 3 Medium – 2 Low – 1

Course Structure

Sl. No	Modules	No. of Lecture Hours	No. of Tutorial Hours	No. of Practical Hours
	Module – 1			
1.1	**Introduction to Strategic Games:** What is game theory? The theory of rational choice	1	-	-
1.2	Strategic games, Examples: The prisoner's dilemma, Bach or Stravinsky	1	1	-
1.3	Matching pennies;	1	-	-
1.4	Nash equilibrium; Examples of Nash equilibrium;	1	1	-
1.5	Best response functions; Dominated actions.	1	1	-
	Module – 2			
2.1	Introduction; Strategic games in which players may randomize	1	1	-
2.2	Mixed strategy Nash equilibrium, Dominated actions	1	-	-
2.3	Pure equilibrium when randomization is allowed	1	1	-
2.4	Illustration: Expert Diagnosis	1	-	-
2.5	Equilibrium in a single population.	1	1	-
	Module – 3			
3.1	Extensive games with perfect information	1	-	-
3.2	Strategies and outcomes	1	1	-
3.3	Nash equilibrium	1	1	-
3.4	Sub- game perfect equilibrium, Finding sub-game perfect equilibria of finite horizon games	1	-	-
3.5	Backward induction, The ultimatum game, Stackelberg's model of duopoly	1	1	-
	Module – 4: -			
4.1	Bayesian Games, Motivational examples	1	-	-
4.2	General definitions, Two examples concerning information	1	1	-
4.3	Illustrations: Cournot's duopoly game with imperfect information	1	1	-
4.4	Providing a public good	1	1	-
4.5	Auctions: Auctions with an arbitrary distribution of valuations	1	-	-
	Module – 5			
5.1	**Competative Games:** Strictly competitive games and maximization	1	-	-
5.2	**Repeated games:** The main idea; Preferences	1	1	-

5.3	Strategies in an infinitely repeated Prisoner's dilemma	1	1	-
5.4	Nash equilibrium of an infinitely repeated Prisoner's dilemma	1	1	-
5.5	Nash equilibrium payoffs of an infinitely repeated Prisoner's dilemma.	1	-	-
	Total No. of Lecture Hours	**25**	-	-
	Total No. of Tutorial Hours		**15**	-
	Total No. of Practical Hours			**00**

Textbook

1. Martin Osborne: "An Introduction to Game Theory", Oxford University Press, First Indian Edition, 2009, 7th impression, ISBN – 0195128958.

Reference books

1. Roger B. Myerson: "Analysis of Conflict Game Theory", Re-print Edition, Harvard University Press, 2008, ISBN – 978-0674341166.
2. Frederick S. Hillier and Gerald J. Lieberman: "Introduction to Operations Research, Concepts and Cases", 9th Edition; Tata McGraw Hill, 2010, ISBN – 0073376299.
3. Joel Watson: "An Introduction to Game Theory" Strategy, 2nd Edition, W.W. Norton &Company, 2007, ISBN – 9780393929348.

Online Resources:

- http://nptel.ac.in/courses.php?disciplineID=111
- http://www.class-central.com/subject/math(Mhttp://nptel.ac.in/courses.php?disciplineID=111
- http://www.class-central.com/subject/math(MOOCs)
- http://academicearth.org/
- VTU e-Shikshana Program
- VTU EDUSAT Program.OOCs)

Code: BDS456A **Course: Scala**

Credits: 01 **L:T:P - 0:0:2**

SEE: NA **CIE: 50 Marks**

SEE Hours: NA **Max. Marks: 50**

Prerequisites if any	NIL
Learning objectives	● Model data using algebraic data types, represented in Scala as families of sealed traits and case classes. ● Use structural recursion and pattern matching to traverse and transform data ● Learn programming with the common data structures of Scala ● Learn object-oriented programming in Scala

Course Outcomes:

On the successful completion of the course, the student will be able to

COs	Course Outcomes	Bloom's level
CO1	Illustrate the Scala syntax and object-oriented principles	Apply
CO2	Interpret - loops, expressions, inheritance, pattern matching	Apply
CO3	write clean and functional Scala codes and test it	Apply
CO4	Illustrate functional programming using Scala	Apply

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12		PSO1	PSO2
CO1	3	3	2	-	3	-	-	-	-	-	3	3		2	3
CO2	2	3	-	2	-	3	-	-	-	-	3	-		3	-
CO3	3	2	-	3	-	-	-	-	-	-	-	-		2	2
CO4	3	3	3	-	2	-	-	-	-	-	-	-		3	-

Mapping Strength: Strong– 3 Medium – 2 Low – 1

List of Experiments

Sl. No.	CO's	Experiments	No. of Practical Hours
1.	CO1	a. Write a Scala program to compute the sum of the two given integer values. If the two values are the same, then return triples their sum. b. Write a Scala program to check two given integers, and return true if one of them is 22 or if their sum is 32.	2
2.	CO1	a. Write a Scala program to remove the character in a given position of a given string. The given position will be in the range 0...string length -1 inclusive. b. Write a Scala program to create a new string taking the first 5 characters of a given string and return the string with the 5 characters added at both the front and back.	2
3.	CO2	a. Write a Scala program to print the multiplication table of a given number using a for loop. b. Write a Scala program to find the largest element in an array using pattern matching	2
4.	CO2	a. Write a Scala function to calculate the product of digits in a given number b. Write a Scala function to check if a given number is a perfect square	2
5.	CO 2	a. Write a Scala program that creates a subclass Student that extends the Person class. Add a property called grade and implement methods to get and set it. b. Write a Scala program that creates a class Triangle with properties side1, side2, and side3. Implement amethod isEquilateral to check if the triangle is equilateral.	2
6.	CO3	a. Write a Scala program that creates an enum class Color with values for different colors. Use the enum class to represent an object's color. b. Write a Scala program that creates a class ContactInfo with properties name, email, and address. Create a class Customer that includes a ContactInfo object.	2
7.	CO3	a. Write a Scala program to create a set and find the difference and intersection between two sets. b. Write a Scala program to create a set and find the second largest element in the set.	2
8.	CO4	a. Write a Scala program to create a list in different ways. Note: Use Lisp style, Java style, Range list, Uniform list, Tabulate list b. Write a Scala program to flatten a given List of Lists, nested list structure.	2
9.	CO4	a. Write a Scala program to add each element n times to a given list of integers. b. Write a Scala program to split a given list into two lists.	1
10	CO4	a. Write a Scala program to swap the elements of a tuple Further print no swapping required if elements are same. b. Write a Scala program to find non-unique elements in a tuple	1
		Total number of practical hours	**18**

Textbook:

1. Programming Scala, Third Edition, O'Reilly Media.
2. Paul Chiusano, Rúnar Bjarnason, Functional Programming in Scala 1st Edition, Manning Publications

Online resources:

- https://docs.scala-lang.org/tutorials/scala-for-java-programmers.html
- https://www.javatpoint.com/scala-tutorial

Code: BDS456B **Course: MongoDB**

Credits: 01 **L:T:P - 0:0:2**

SEE: NA **CIE: 50 Marks**

SEE Hours: NA **Max. Marks: 50**

Prerequisites if any	NIL
Learning objectives	● Understand basic MongoDB functions, operators and types of operations in MongoDB. ● Demonstrate the use of Indexing, Advanced Indexing in MongoDB. ● Apply the aggregation and Map Reduction in MongoDB. ● Demonstrate text searching on collections in MongoDB.

Course Outcomes:

On the successful completion of the course, the student will be able to

COs	Course Outcomes	Bloom's level
CO1	Use MangoDB commands and queries.	Apply
CO2	Illustrate the role of aggregate pipelines to extract data.	Apply
CO3	Demonstrate optimization of queries by creating indexes.	Apply
CO4	Develop aggregate pipelines for text search in collections.	Apply

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12		PSO1	PSO2
CO1	3	3	2	-	3	-	-	-	-	-	3	3		3	3
CO2	3	2	-	2	-	2	-	-	-	-	3	-		2	-
CO3	3	2	-	2	-	-	-	-	-	-	-	-		2	2
CO4	3	3	3	-	3	-	-	-	-	-	-	-		3	-

Mapping Strength: **Strong– 3** **Medium – 2** **Low – 1**

List of Experiments

Sl. No.	CO's	Experiments	No. of Practical Hours
1.	CO1	a) Illustration of Where Clause, AND,OR operations in MongoDB. b) Execute the Commands of MongoDB and operations in MongoDB : Insert, Query, Update, Delete and Projection. (Note: use any collection) [Refer: Book 1 chapter 4].	2
2.	CO1	a. Develop a MongoDB query to select certain fields and ignore some fields of the documents from any collection. b. Develop a MongoDB query to display the first 5 documents from the results obtained in a. [use of limit and find] [Refe: Book1 Chapter 4, book 2: chapter 5]	2

3.	CO2	a. Execute query selectors (comparison selectors, logical selectors) and list out the results on any collection b. Execute query selectors (Geospatial selectors, Bitwise selectors) and list out the results on any collection [Refer: Book 3 Chapter 13]	2
4.	CO2	Create and demonstrate how projection operators ($, $elematch and $slice) would be used in the MondoDB. [Refer: Book 3 Chapter 14]	2
5.	CO 2	Execute Aggregation operations ($avg, $min,$max, $push, $addToSet etc.). students encourage to execute several queries to demonstrate various aggregation operators) [Refer: Book 3 Chapter 15]	2
6.	CO3	Execute Aggregation Pipeline and its operations (pipeline must contain $match, $group, $sort, $project, $skip etc. students encourage to execute several queries to demonstrate various aggregation operators) [refer book 2: chapter 6]	2
7.	CO3	a. Find all listings with listing_url, name, address, host_picture_url in the listings And Reviews collection that have a host with a picture url b. Using E-commerce collection write a query to display reviews summary. [refer Book2: chapter 6].	2
8.	CO4	a. Find all listings with listing_url, name, address, host_picture_url in the listings And Reviews collection that have a host with a picture url b. Using E-commerce collection write a query to display reviews summary. [refer Book2: chapter 6]	2
9.	CO4	a. Develop a query to demonstrate Text search using catalog data collection for a given word b. Develop queries to illustrate excluding documents with certain words and phrases Refer: Book 2: Chapter 9]	1
10	CO4	Develop an aggregation pipeline to illustrate Text search on Catalog data collection. Refer: Book 2 :Chapter 9]	1
		Total number of practical hours	**18**

Textbook:

1. **“**MongoDB: The Definitive Guide”, Kristina chodorow, 2nd ed O’REILLY, 2013.
2. **“**MongoDB in Action” by KYLE BANKER et. al. 2nd ed, Manning publication, 2016
3. “MongoDB Complete Guide” by Manu Sharma 1st ed, bpb publication, 2023.

Online resources:

- https://www.youtube.com/watch?v=dEm2AS5amyA
- https://www.w3resource.com/mongodb-exercises/

Code: BDS456C **Course: MERN (MongoDB, Express, React, Node)**

Credits: 1 **L: T: P – 0:0:2**

SEE: NA **CIE: 50 Marks**

SEE Hours:NA **Max. Marks: 50**

Prerequisites if any	Fundamentals of Web development and Client server technology
Learning objectives	• To gain proficiency in applying advanced Internet technology scripting languages, tools used in client server side for understanding, analyzing and applying. • To gain a comprehensive understanding of different internet technology tools like Express, React, Node and MongoDB which enables students to analyze, compare and apply them effectively in various problem solving contexts

Course Outcomes: On the successful completion of the course, the student will be able to

COs	Course Outcomes	Bloom's level
CO1	Apply the fundamentals of MongoDB, such as data modelling, CRUD operations, and basic queries to solvegiven problem.	Understanding
CO2	Use constructs of Express.js, including routing, software and constructing RESTful APIs to solve real worldproblems	Apply
CO3	Develop scalable and efficient RESTful APIs using NodeJS.	Apply
CO4	Develop applications using React, including components, state, props, and JSX syntax	Analyze

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12		PSO1	PSO2
CO1	2	2	-	-	2	-	2	2	-	-	-	3		-	2
CO2	2	2	-	-	2	-	2	2	-	-	-	3		-	2
CO3	2	2	-	-	2	-	2	2	-	-	-	3		-	2
CO4	2	2	-	-	2	-	2	2	-	-	-	3		-	2

Mapping Strength: Strong– 3 Medium – 2 Low – 1

Course Structure

SL. No.	CO	Experiments	No. of Practical Hours
1	CO1	a.Using MongoDB, create a collection called transactions in database usermanaged (drop if it already exists) and bulk load the data from a json file, transactions.json b. Upsert the record from the new file called transactions_upsert.json in Mongodb.	2
2	CO1	Query MongoDB with Conditions: [Create appropriate collection with necessary documents to answer the query] a. Find any record where Name is Somu b. Find any record where total payment amount (Payment.Total) is 600. c. Find any record where price (Transaction.price) is between 300 to 500. d. Calculate the total transaction amount by adding up Payment.Total in all records.	2
3	CO1	Implement all CRUD operations on a File System using Node JS	1
4	CO2	a. Write a program to check request header for cookies. b. write node.js program to print the a car object properties, delete the second property and get length of the object.	2
5	CO2	Develop an authentication mechanism with email_id and password using HTML and Express JS (POST method)	2
6	CO2	Develop two routes: find_prime_100 and find_cube_100 which prints prime numbers less than 100 and cubes less than 100 using Express JS routing mechanism	2
7	CO1	. Read the data of a student containing usn, name, sem, year_of_admission from node js and store it in the MongoDB . For a partial name given in node js, search all the names from MongoDB student documents created in Question(a)	2
8	CO3	Develop the application that sends fruit name and price data from client side to Node.js server using Ajax	2
9	CO4	Develop a React code to build a simple search filter functionality to display a filtered list based on the search query entered by the user.	2
10	CO4	Develop a React code to collect data from rest API.	1
		Total No. of Practical Hours	**18**

Textbook:

1. Vasan SubramanianPro MERN Stack: Full Stack Web App Development with Mongo, Express, React, and Node, Apress; 1st ed. edition (1 April 2017)
2. Eddy Wilson Iriarte Koroliova,MERN Quick Start Guide, Packt Publishing (31 May 2018)

Online Resources:

1. https://www.geeksforgeeks.org/mern-stack/
2. https://blog.logrocket.com/mern-stack-tutorial/

Code: BDS456D **Course: Technical Writing using LaTeX**

Credits: 1 **L: T: P – 0:0:2**

SEE: NA **CIE: 50 Marks**

SEE Hours:NA **Max. Marks: 50**

Prerequisites if any	Fundamentals of Web development and Client server technology
Learning objectives	● To introduce the basic syntax and semantics of the LaTeX scripting language ● To understand the presentation of tables and figures in the document ● To illustrate the LaTeX syntax to represent the theorems and mathematical equations ● To make use of the libraries (Tikz, algorithm) to design the diagram and algorithms in the document

Course Outcomes:

On the successful completion of the course, the student will be able to

COs	Course Outcomes	Bloom's level
CO1	Explain the basic LaTeX command to develop simple document	Understand
CO2	Develop LaTeX script to present the tables, figures, theorems, and mathematical equations in the document	Apply
CO3	Develop programs to generate the complete report with citations and a bibliography	Apply
CO4	Illustrate the use of Tikz and algorithm libraries to design graphics and algorithms in the document	Analyze

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12		PSO1	PSO2
CO1	2	-	-	3	-	-	-	-	-	3	-	3		-	3
CO2	2		-	-	-	-	-	-	-	3	-	2		-	2
CO3	2	2	-	3	-	-	-	-	-	-	-	-		-	2
CO4	3	3	-	-	-	-	-	-	-	-	-	-		-	3

Mapping Strength: Strong– 3 Medium – 2 Low – 1

Course Structure

<table>
<tr><th>SL. No.</th><th>CO</th><th>Experiments</th><th>No. of Practical Hours</th></tr>
<tr><td>1</td><td>CO1</td><td>Develop a LaTeX script to create a simple document that consists of 2 sections [Section1, Section2], and a paragraph with dummy text in each section. And also include header [title of document] and footer [institute name, page number] in the document.</td><td>2</td></tr>
<tr><td>2</td><td>CO1</td><td>Develop a LaTeX script to create a document that displays the sample Abstract/Summary</td><td>2</td></tr>
<tr><td>3</td><td>CO1</td><td>Develop a LaTeX script to create a simple title page of the VTU project Report [Use suitable Logos and text formatting]</td><td>1</td></tr>
<tr><td>4</td><td>CO2</td><td>Develop a LaTeX script to create the Certificate Page of the Report [Use suitable commands to leave the blank spaces for user entry]</td><td>2</td></tr>
<tr><td>5</td><td>CO2</td><td>Develop a LaTeX script to create a document that contains the following table with proper labels.
<table>
<tr><th rowspan="2">S.No</th><th rowspan="2">USN</th><th rowspan="2">Student Name</th><th colspan="3">Marks</th></tr>
<tr><th>Subject1</th><th>Subject2</th><th>Subject3</th></tr>
<tr><td>1</td><td>4XX22XX001</td><td>Name 1</td><td>89</td><td>60</td><td>90</td></tr>
<tr><td>2</td><td>4XX22XX002</td><td>Name 2</td><td>78</td><td>45</td><td>98</td></tr>
<tr><td>3</td><td>4XX22XX003</td><td>Name 3</td><td>67</td><td>55</td><td>59</td></tr>
</table></td><td>2</td></tr>
<tr><td>6</td><td>CO2</td><td>Develop a LaTeX script to include the side-by-side graphics/pictures/figures in the document by using the subgraph concept</td><td>2</td></tr>
<tr><td>7</td><td>CO1</td><td>Develop a LaTeX script to create a document that consists of the following two mathematical equations
$x = \frac{-b \pm \sqrt{b^2-4ac}}{2a}$
$= \frac{-2 \pm \sqrt{2^2-4*(1)*(-8)}}{2*1}$
$= \frac{-2 \pm \sqrt{4+32}}{2}$
$\varphi_\sigma^\lambda A_t = \sum_{\pi \in C_t} \operatorname{sgn}(\pi) \varphi_\sigma^\lambda \varphi_\pi^\lambda$
$= \sum_{\tau \in C_{\sigma t}} \operatorname{sgn}(\sigma^{-1}\tau\sigma) \varphi_\sigma^\lambda \varphi_{\sigma^{-1}\tau\sigma}^\lambda$
$= A_{\sigma t} \varphi_\sigma^\lambda$</td><td>2</td></tr>
<tr><td>8</td><td>CO3</td><td>Develop a LaTeX script to demonstrate the presentation of Numbered theorems, definitions, corollaries, and lemmas in the document</td><td>2</td></tr>
<tr><td>9</td><td>CO4</td><td>Develop a LaTeX script to create a document that consists of two paragraphs with a minimum of 10 citations in it and display the reference in the section</td><td>2</td></tr>
<tr><td>10</td><td>CO4</td><td>Develop a LaTeX script to design a simple tree diagram or hierarchical structure in the document with appropriate labels using the Tikz library</td><td>1</td></tr>
<tr><td colspan="3">Total No. of Practical Hours</td><td>18</td></tr>
</table>

Textbook:

1. A Short Introduction to LaTeX BY FIRUZA KARMALI (AIBARA), A book for beginners, 2019
2. Formatting Information: A Beginner's Introduction to Typesetting with LaTeX, BY PETER FLYNN, Comprehensive TeX Archive Network (2005)

Online Resources:

- LaTeX TUTORIAL: [https://latex-tutorial.com/tutorials\
- LaTeX TUTORIAL: [https://www.javatpoint.com/latex

Course Code: BBOC407 **Course: Biology for Engineers**

Credits: 2 **L:T:P 2:0:0**

SEE: 50 Marks **CIE: 50 Marks**

SEE Hours: 2 Hrs **Max. Mark:100**

Prerequisites if any	None
Learning objectives	1. Review the basics of cell biology and role of biomolecules. 2. Elucidate the significance of Biomechanics and Biomaterials

Course Outcomes:

On the successful completion of the course, the student will be able to

	Course Outcomes	Bloom's level
CO1	Explain the fundamentals of Life, Evolution, Biomolecules, Cell Biology & Genetics	Understand
CO2	Outline the physical principles regulating the systems of the human bodies	Understand
CO3	Describe the impact of Biomaterials on the fields of Engineering & Medicine	Understand
CO4	Summarize tissue engineering and clinical applications of materials	Understand

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12		PSO1	PSO2
CO1	3	-	-	-	-	3	2	-	-	-	-	3		-	-
CO2	3	-	-	-	-	3	2	-	-	-	-	3		-	-
CO3	3	-	-	-	-	3	2	-	-	-	-	3		-	-
CO4	3	-	-	-	-	3	2	-	-	-	-	3		-	-

Strong: 3 Medium: 2 Low: 1

Course Structure

Sl. No.	Modules	No. of Lecture Hours	No. of Tutorial Hours	Self-Learning Hours
	Module - 1			
1.1	**Cell basic unit of life:** Introduction, Origin, and evaluation of life. Structure and functions of a cell. Stem cells and their application.	2	-	-
1.2	Biomolecules: Properties and functions of Carbohydrates, Nucleic acids, proteins, lipids. Importance of special biomolecules: Properties and functions of enzymes, vitamins, and hormones.	3	-	-
	Module - 2			
2.1	**Application of biomolecules:** Carbohydrates in cellulose-based water filters production, PHA and PLA in bioplastics production, Nucleic acids in vaccines and diagnosis,	2	-	-
2.2	Proteins in food production, lipids in biodiesel and detergents production, Enzymes in biosensors fabrication, food processing, detergent formulation, and textile processing.	3	-	-
	Module - 3			
3.1	**Adaptation of anatomical principles for bioengineering design:** Brain as a CPU system, Eye as a Camera system, Heart as a pump system.	3	-	-
3.2	Lungs as purification system, Kidney as a filtration system.	2	-	-
	Module - 4			
4.1	**Nature-bioinspired materials and mechanisms:** Echolocation, Photosynthesis, Bird flying, Lotus leaf effect, Plant burrs, Shark skin, Kingfisher beak.	3	-	-
4.2	Human Blood substitutes - haemoglobin-based oxygen carriers (HBOCs) and perfluorocarbons (PFCs).	2	-	-
	Module - 5			
5.1	**Trends in bioengineering:** Muscular and Skeletal Systems as scaffolds, and tissue engineering, Bioprinting techniques and materials.	3	-	-
5.2	Electrical tongue and electrical nose in food science, DNA origami and Biocomputing, Bioimaging and Artificial Intelligence for disease diagnosis, Bio concrete, Bioremediation, Biomining.	2	-	-
	Total No. of Lecture Hours	25	-	-
	Total No. of Tutorial Hours		15	-
	Total No. of Self learning Hours -			00

Text Books:

1. Biology for Engineers by G. K. Suraishkumar; Oxford University Press, 2019, First Edition

Reference Books:

1. Introductory Biomechanics: From Cells to Organisms by C. Ross Ethier and Craig A. Simmons; Cambridge University Press, 2012, Online Edition
2. Introduction to Biomaterials: Basic Theory with Engineering Applications, J. L. Ong, Mark R. Appleford, Gopinath Mani, Cambridge University Press, 2014, First Edition
3. Biology for Engineers, Rajendra Singh C and Rathnakar Rao N, Rajendra Singh C and Rathnakar Rao, N Publishing, Bengaluru, 2023.
4. Human Physiology, Stuart Fox, Krista Rompolski, McGraw-Hill eBook. 16th Edition, 2022
5. Biology for Engineers, Thyagarajan S., Selvamurugan N., Rajesh M.P., Nazeer R.A., Thilagaraj W., Barathi S., and Jaganthan M.K., Tata McGraw-Hill, New Delhi, 2012.
6. Biology for Engineers, Arthur T. Johnson, CRC Press, Taylor and Francis, 2011
7. Biomedical Instrumentation, Leslie Cromwell, Prentice Hall 2011.
8. Biology for Engineers, Sohini Singh and Tanu Allen, Vayu Education of India, New Delhi, 2014.
9. Biomimetics: Nature-Based Innovation, Yoseph Bar-Cohen, 1st edition, 2012, CRC Press.
10. Bio-Inspired Artificial Intelligence: Theories, Methods and Technologies, D. Floreano and C. Mattiussi, MIT Press, 2008.
11. Bioremediation of heavy metals: bacterial participation, by C R Sunilkumar, N Geetha A C Udayashankar Lambert Academic Publishing, 2019.
12. 3D Bioprinting: Fundamentals, Principles and Applications by Ibrahim Ozbolat, Academic Press, 2016.
13. Electronic Noses and Tongues in Food Science, Maria Rodriguez Mende, Academic Press, 2016

Online Resources:

1. NOC: Biology for engineers and other non-biologists, IIT Madras; Dr. Madhulika Dixit, Prof. G.K. Suraishkumar, https://nptel.ac.in/courses/121106008
2. Introduction To Biological Engineering Design, MIT Open Courseware, https://ocw.mit.edu/courses/20-020-introduction-to-biological-engineering-design-spring-2009
3. Introduction To Bioengineering, MIT Open Courseware, https://ocw.mit.edu/courses/20-010j- introduction-to-bioengineering-be-010j-spring-2006

Course Code: BUHK408 **Course Name: Universal Human Values and Professional Ethics**

Credits: 1 **L:T:P - 1:0:0**

SEE Marks : 50 **CIE : 50 Marks**

SEE Duration: 2 Hours

Prerequisites if any	Nil
Learning objectives	1. To help the students appreciate the essential complementarily between 'VALUES' and 'SKILLS' to ensure sustained happiness and prosperity, which are the core aspirations of all human beings. 2. To facilitate the development of a Holistic perspective among students towards life and profession as well as towards happiness and prosperity based on a correct understanding of the Human reality and the rest of existence. 3. To highlight plausible implications of such a Holistic understanding in terms of ethical human conduct, trustful and mutually fulfilling human behavior and mutually enriching interaction with nature

Course Outcomes:

On the successful completion of the course, the student will be able to

COs	Course Outcomes	Bloom's level
CO1	Explain the fundamental aspirations of human beings and the importance of a fulfilling life	Understand
CO2	Develop an understanding of universal human values and apply them for value-based living.	Apply
CO3	Demonstrate responsible human behavior and nurture mutually enriching relationships with nature.	Apply

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12		PSO1	PSO2
CO1	-	-	-	-	-	3	2	2	-	-	-	2		-	-
CO2	-	-	-	-	-	2	3	2	2	2	-	3		-	-
CO3	-	-	-	-	-	3	3	2	2	-	-	3		-	-

S – Strong (3) M – Medium (2) L – Low (1)

Course Structure

Sl. No.	Modules	No. of Lecture Hours	No. of Tutorial Hours
Module – 1: Introduction - Need, Basic Guidelines, Content and Process for Value Education			
1.1	Understanding the need, basic guidelines, content and process for Value Education	1	Nil
1.2	Self-Exploration–what is it? - its content and process; 'Natural Acceptance' and Experiential Validation- as the mechanism for self-exploration	1	Nil
1.3	Right understanding, Relationship and Physical Facilities- the basic requirements for fulfillment of aspirations of every human being with their correct priority	1	Nil
1.4	Method to fulfill the above human aspirations: understanding and living in harmony at various levels . Practice session	2	Nil
Module – 2: Understanding Harmony in Myself, Family, Society and Human Relationship			
2.1	Understanding human being as a co-existence of the sentient 'I' and the material 'Body', Understanding the needs of Self ('I') and 'Body' - Sukh and Suvidha	1	Nil
2.2	Understanding Harmony in the family – the basic unit of human interaction. Understanding values in human-human relationship; meaning of Nyaya and program for its fulfillment to ensure Ubhay- tripti; Trust (Vishwas) and Respect (Samman) as the foundational values of relationship	1	Nil
2.3	Understanding the meaning of Vishwas and Samman; Difference between intention and competence; respect and differentiation ; Understanding the harmony in the society (society being an extension of family): Samadhan, Samridhi, Abhay, Sah-astitva as comprehensive Human Goals	1	Nil
2.4	Visualizing a universal harmonious order in society- Undivided Society (Akhand Samaj), Universal Order (Sarvabhaum Vyawastha)- from family to world family! Practice session	2	Nil
Module – 3: Understanding Harmony in the Nature, Existence and Implications of the all Holistic on Professional Ethics			
3.1	Understanding the harmony in the Nature, Interconnectedness and mutual fulfillment among the four orders of nature recyclability and self-regulation in nature	1	Nil
3.2	Understanding Existence as Co-existence (Sah-astitva) of mutually interacting units in all-pervasive space.	1	Nil
3.3	Competence in professional ethics: a) Ability to utilize the professional competence for augmenting universal human order b) Ability to identify and develop appropriate technologies and management patterns for above production systems.	1	Nil

3.4	Strategy for transition from the present state to Universal Human Order: a) At the level of individual: as socially and ecologically responsible engineers, technologists and managers b) At the level of society: as mutually enriching institutions and organizations. Practice session	2	Nil
	Total No. of Lecture Hours	15	
	Total No. of Tutorial Hours		Nil

Guidelines and Content for Practice Sessions

Practice Session 1:

Introduce yourself in detail. What are the goals in your life? How do you set your goals in your life? How do you differentiate between right and wrong? What have been your achievements and shortcomings in your life? Observe and analyze them.

Expected outcome: The students start exploring themselves; get comfortable to each other and to the teacher and start finding the need and relevance for the course.

Practice Sessions 2:

1. a. Observe that any physical facility you use, follows the given sequence with time: Necessary & tasteful→ unnecessary & tasteful → unnecessary & tasteless → intolerable b. In contrast, observe that any feeling in you is either naturally acceptable or not acceptable at all. If naturally acceptable, you want it continuously and if not acceptable, you do not want it any moment!

2. List down all your activities. Observe whether the activity is of 'I' or of Body or with the participation of both 'I' and Body.

Expected outcome:

1. The students are able to see that all physical facilities they use are required for a limited time in a limited quantity. Also they are able to see that in case of feelings, they want continuity of the naturally acceptable feelings and they do not want feelings which are not naturally acceptable even for a single moment.

2. The students are able to see that activities like understanding, desire, thought and selection are the activities of 'I' only, the activities like breathing, palpitation of different parts of the body are fully the activities of the body with the acceptance of 'I' while the activities they do with their sense organs like hearing through ears, seeing througbody.es, sensing through touch, tasting through tongue and smelling through nose or the activities they do with their work organs like hands, legs etc. are such activities that require the participation of both 'I' and body

Practice Session 3:

Form small groups in the class and in that group initiate dialogue and ask the eight questions related to trust. The eight questions are:

1a. Do I want to make myself happy?

2a. Do I want to make others happy?

3a. Does the other want to make him happy?

4a. Does the other want to make me happy?

What is the answer? Intention (Natural Acceptance) 1b. Am I able to make myself always happy?

2b. Am I able to make others happy?

3b. Is the other able to make him always happy? 4b. Is the other able to make me always happy?

What is the answer? Competence

Let each student answer the questions for himself and everyone else. Discuss the difference between intention and competence. Observe whether you evaluate your intention & competence as well as the others' intention & competence

Expected outcome:

The students are able to see that the first four questions are related to our Natural Acceptance i.e. Intention and the next four to our Competence. They are able to note that the intention is always correct, only competence is lacking! We generally evaluate ourselves on the basis of our intention and others on the basis of their competence! We seldom look at our competence and others' intention as a result we conclude that I am a good person and other is a bad person.

Textbooks:

1. R.R Gaur, R Sangal, G P Bagaria, A foundation course in Human Values and professional Ethics, Excel books, New Delhi, 2010, ISBN 978-8-174-46781-2.

References:

1. IIT Delhi, Modern Technology – the Untold Story

Course Code: BNSK459 **Course: National Service Scheme (NSS)**

Credits: Zero **L:T:P 0:0:2**

SEE: NA **CIE: 100 Marks**

SEE Hours: NA **Max. Marks: 100**

Prerequisites if any	1. Students should have a service oriented mind set and social concern. 2. Students should have dedication to work at any remote place, anytime with available resources and proper time management for the other works. 3. Students should be ready to sacrifice some of the time and wishes to achieve service oriented targets on time
Learning objectives	1. Understand the community in which they work 2. Identify the needs and problems of the community and involve them in problem- solving 3. Develop among themselves a sense of social & civic responsibility & utilize their knowledge in finding practical solutions to individual and community problems 4. Develop competence required for group-living and sharing of responsibilities & gain skills in mobilizing community participation to acquire leadership qualities and democratic attitudes 5. Develop capacity to meet emergencies and natural disasters & practice national integration and social harmony

Course Outcomes: *On the successful completion of the course, the student will be able to*

COs	Course Outcomes
CO1	Understand the importance of his / her responsibilities towards society.
CO2	Analyse the environmental and societal problems/issues and will be able to design solutions for the same
CO3	Evaluate the existing system and to propose practical solutions for the same for Sustainable development and Implement government or self-driven projects effectively in the field.

Mapping With POs and PSOs

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12		PSO1	PSO2	PSO3	PSO4
CO1	-	-	-	-	-	3	2	2	-	2	-	2		To be identified for each branch by Course Instructor			
CO2	-	-	-	-	-	3	2	2	-	2	-	2					
CO3	-	-	-	-	-	3	3	2	-	2	-	2					

Mapping Strength: 2 **Strong: 3** **Medium –2** **Low – 1**

Course Structure

SI.No	Modules	No. of Lecture Hours	No. of Tutorial Hours	No. of Practical Hours
	List of Events			
1	Organic farming, Indian Agriculture (Past, Present and Future) Connectivity for marketing.			
2	Waste management– Public, Private and Govt organization, 5 R's.			
3	Setting of the information imparting club for women leading to contribution in social and economic issues.			
4	Water conservation techniques – Role of different stakeholders– Implementation.			
5	Preparing an actionable business proposal for enhancing the village income and approach for implementation.			
6	Helping local schools to achieve good results and enhance their enrolment in Higher/ technical/ vocational education.			
7	Developing Sustainable Water management system for rural areas and implementation approaches.			
8	Contribution to any national level initiative of Government of India. Digital India, Skill India, Swachh Bharat, Atmanirbhar Bharath, Make in India, Mudra scheme, Skill development programs etc.			
9	Spreading public awareness under rural outreach programs.(minimum5 programs)			
10	Social connect and responsibilities.			
11	Plantation and adoption of plants.			
12	Govt. school Rejuvenation and helping them to achieve good infrastructure			
13	Sustainable agriculture practices - Organic farming, Agroforestry and crop rotation.			
14	Rural finance – role of local bodies, need for agricultural finance and sources of agricultural finances.			
14	Rural finance – role of local bodies, need for agricultural finance and sources of agricultural finances.			
15	Strategies for the development of rural markets and emerging issues in rural marketing			
16	Rural energy system – conventional and non-conventional, Rural electrification- policies, achievements and targets.			
17	Livestock economies - fishery and poultry development, forestry and horticulture.			
18	Role of NGO's in rural development, the role of voluntary organization in India's development processes.			

19	Issues in rural industrialization and development of agro-based industries, rural non-farm sector			
Total No. of Lecture Hours		-	-	-
Total No. of Tutorial Hours			-	-
Total No. of Practical Hours				**26**

ONENSS – CAMP @ College /University /State or Central Govt Level /NGO's /General Social Camps

- Students have to take up anyone activity on the above said topics and have to prepare content for awareness and technical contents for implementation of the projects and have to present strategies for implementation of the same. Compulsorily students have to attend one camp.
- CIE will be evaluated based on their presentation, approach and implementation strategies.

Suggested Learning Resource:

NSS Course Manual, Published by NSS Cell, VTU Belagavi.

Course Code: BPEK459 **Course: Physical Education (Sports & Athletics) – II**

Credits: Zero **L:T:P 0:0:2**

SEE: NA **CIE: 100 Marks**

SEE Hours: NA **Max. Marks: 100**

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

1. Understand the ethics and moral values in sports and athletics
2. Perform in the selected sports or athletics of student's choice.
3. Understand the roles and responsibilities of organisation and administration of sports and games.

Module IV : Ethics and Moral Values **5 Hours**
A. Ethics in Sports B. Moral Values in Sports and Games
Module V : Specific Games (Any one to be selected by the student) **20 Hours** A. Volleyball – Attack, Block, Service, Upper Hand Pass and Lower hand Pass. B. Throwball – Service, Receive, Spin attack, Net Drop & Jump throw. C. Kabaddi – Hand touch, Toe Touch, Thigh Hold, Ankle hold and Bonus. D. Kho-Kho – Giving Kho, Single Chain, Pole dive, Pole turning, 3-6 Up. E. Table Tennis – Service (Fore Hand & Back Hand), Receive (Fore Hand & Back Hand), Smash. F. Athletics (Track / Field Events) – Any event as per availability of Ground.
Module VI: Role of Organisation and administration **5 Hours**

Scheme and Assessment for auditing the course and Grades:

Sl. No.	Activity	Marks
1.	Participation of student in all the modules	20
2.	Quizzes – 2, each of 15 marks	30
3.	Assignment	50
	Total	**100**

Course Code: BYOK459 **Course: Yoga**

Credits: Zero **L:T:P 0:0:2**

SEE: NA **CIE: 100 Marks**

SEE Hours: NA **Max. Marks: 100**

Prerequisites if any	None
Learning objectives	1. To enable the student to have good health and mental hygiene. 2. To possess emotional stability 3. To integrate moral values 4. To attain higher level of consciousness.

Course Outcomes:

On successful completion of the course, the student will be able to:

Course Outcomes		Bloom's level
CO1	Understand the meaning of Yoga, its origin, history, development and importance.	Understand
CO2	Perform various Surya namaskar and able to Teach its benefits	Apply
CO3	Perform various asanas and able to Teach its benefits	Apply
CO4	Understand Benefits of Yoga on fitness and health	Apply

Course Content

SI.No	Module – 1	No. of Lecture and Practical Hours	No. of Tutorial Sessions
1.1	Role of yoga in controlling diseases	1	-
1.2	Patanjali's Ashtanga Yoga, its need and importance.	1	-
1.3	Yama :Ahimsa, satya, asteya, brahmacarya, aparigraha	1	-
1.4	Niyama :shoucha, santosh, tapa, svaadhyaya, Eshvarapranidhan	1	-
	Module – 2		
2.1	Warmup Exercise	2	-
2.2	Yoga jogging	2	-
2.3	Suryanamaskar12 count- 4 rounds of practice	2	-
2.4	Asana its meaning by name, technique, precautionary measures and benefits of each asana.	2	-
2.5	Sitting: 1.Sukhasana 2. Paschimottanasana 3.Bharadwajasana	2	-
2.6	Standing: 1. Ardhakati Chakrasana 2. Parshva Chakrasana	2	-
2.7	Prone line: 1.Makarasana 2.Dhanurasana	2	
2.8	Supine line 1. Halasana 2. Karna Peedasana	2	
	Module – 3		
3.1	Pranayama – Suryanuloma, Chandranuloma,	2	-
3.2	Suryabhedana,Chandra Bhedana,Nadishodhana	2	-
	Total No. of Lecture and practical Hours	**24**	**-**
	No. of Tutorial Sessions		**Nil**

Detailed Lesson Plan

Sl. No. of Module	Number of related learning Objectives	Weeks/ Dates	Online Mode		ICT Tool/ Platform/ LMS	Face-to-face Mode		Duration in Minutes
			Resource (OER/ URL/ IM/ CP)	Activity (Describe activity in detail)		Resource (OER/ URL/ IM/ CP)	Activity	
1.1	1 & 4	W 1					Explanation	60
1.2	1 & 4	W 1					Explanation	60
1.3	1 & 4	W 2					Explanation	60
1.4	1 & 4	W 2					Explanation	60
1.5	1 & 4	W 3					Explanation	60
1.6	1 & 4	W 3					Explanation	60
2.1	2 & 4	W 4					Practicing Surya namaskar	60
2.2	2 & 4	W 4					Practicing Surya namaskar	60
2.3	2 & 4	W 5					Practicing Surya namaskara	60
2.4	2 & 4	W 5					Practicing Surya namaskar	60
2.5	2 & 4	W 6					Practicing Surya namaskar	60
2.6	2 & 4	W 6					Practicing Asana	60
3.1	3 & 4	W 7					Practicing Asana	60
3.2	3 & 4	W 7					Practicing Asana	60
3.3	3 & 4	W 8					Practicing Asana	60
3.4	3 & 4	W 8					Practicing Asana	60
3.5	3 & 4	W 9					Practicing Asana	60
3.6	3 & 4	W 9					Practicing Asana	60
3.7	3 & 4	W10					Practicing Asana	60
3.8	3 & 4	W10					Practicing Asana	60
3.9	3 & 4	W 11					Practicing Asana	60
3.10	3 & 4	W11					Practicing Asana	60
3.11	3 & 4	W12					Practicing Asana	60
3.12	3 & 4	W 12					Practicing Asana	60

Assessment Pattern:

Bloom's level	Continuous Internal Examination			Semester End Examination
	Test 1	Test 2	Assignment	
Remember	-	-	-	-
Understand	✓	✓	✓	✓
Apply	✓	✓	✓	✓

Suggested Learning Resources:

Text Books:

1. Yogapravesha in Kannada by Ajitkumar
2. Light on Yoga by BKS Iyengar
3. Teaching Methods for Yogic practices by Dr. M L Gharote & Dr. S K Ganguly
4. Yoga Instructor Course hand book published by SVYASA University, Bengaluru
5. Yoga for Children –step by step – by Yamini Muthanna

Web links and Video Lectures (e-Resources): Refer links

1. https://youtu.be/KB-TYlgd1wE
2. https://youtu.be/aa-TG0Wg1Ls

Course Code: BMUK359 /459/559/ 658 **Course: Music**

Credits: Zero **L:T:P 0:0:2**

SEE: NA **CIE: 100 Marks**

SEE Hours: NA **Max. Marks: 100**

Prerequisites if any	None
Learning objectives	1. Identify the major traditions of Indian music, both through notations and aurally. 2. Analyse the compositions with respect to musical and lyrical content 3. Demonstrate an ability to use music technology appropriately in a variety of setting.

Course Outcomes:

On successful completion of the course, the student will be able to:

Course Outcomes		Bloom's level
CO1	Discuss the Indian system of music and relate it to other genres (Cognitive Domain)	Understand
CO2	Experience the emotions of composer and develop empathy (Affective Domain)	Apply
CO3	Respond to queries on various patterns in a composition (Psycho Motor Domain)	Apply

Course Content

Sl. No.	Modules	No. of practical and Lecture Hours	No. of Tutorial Hours
1.1	**Preamble:** Contents of the curriculum intend to promote music as language to develop an analytical, Creative, and intuitive Understanding. For this the student must experience music through study and direct participation in improvisation and composition.	2	-
1.2	Origin of the Indian Music: Evolution of the Indian Music system, understanding of Shruthi, Nada, Swara, Laya. Raga Tala, Mela	1	-

2.1	**Compositions:** Introduction to the types of composition in Carnatic Music-Geethe, Jathi, Swara, Swarajathi, Varna, Krithi, and Thillana, Notation System	3	-
3.1	**Composers:** Biography and contributions of Purandaradasa, Thyagaraja, Mysuru Vasudevacharya.	3	-
4.1	**Music Instruments:** Classification and construction of string instruments, wind instruments, percussion instruments, Idiophones (Ghana Vaadya), Examples of each class of Instruments	3	-
5.1	**Abhyasa Gana:** Singing the swara exercises (Sarale Varase Only), Botation writing for Sarale Varase and Suladi Saptha Tala (Only in Mayamalavagowla Raga), Singing 4 Geethe in Malahari, and one jathi Swara, One Krithi in a Mela raga	14	-
Total No. of Lecture and Practical Hours		***26***	
Total No. of Tutorial Hours			**00**

Teaching Practice:

- Classroom Teaching
- ICT - Power Point Presentation
- Audio & Video Visualization Tools

CIE: 100 Marks

- CIE 1 for 40 marks - A theory paper which is MCQ/ Descriptive conducted during the semester
- CIE 2 for 60 marks- A practical test conducted at the semester in which the student has to recite one Sarale Varase mentioned by the examiner in three speeds. Sing / Play the Geethe in Malahari. Singing/Playing Jathi Swara / Krithi.
- Student have to secure minimum of 40% out of (40+60)100 marks for qualifying in this course.

Text Books:

1. Vidushi Vasantha Madhavi, " Theory of Music", Prism Publication, 2007.
2. T Sachidevi and T Sharadha (Thirumalai Sisters), Karnataka Sangeetha Dharpana- Vol. 1 (English), Shreenivaas Prakaashana, 2018.

References:

1. Lakshminarayana Subramaniam, Viji Subramanaim, "Classical Music of India: A Practical Guige", Tranqueber 2018.
2. R. Rangaramanuja Ayyangar, "History of South Indian (Carnatic) Music", Vipanci Charitable Trust; Third edition, 2019.
3. Ethel Rosenthal," The Story of Indian Music and Its Instruments: A Study of the Present and a Record of the Past", Pilgrims Publishing, 2007.
4. Carnatic Music, National Institute of Open Schooling, 2019.