Mahatma Education Society's

Pillai HOC College of Arts, Science & Commerce
(Autonomous)
Rasayani

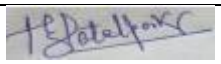
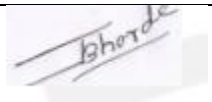
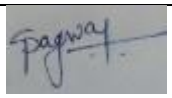
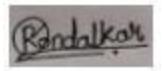
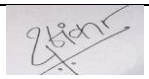
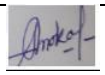
Affiliated to University of Mumbai
NAAC Accredited with "A+" Grade in cycle II
ISO 9001:2015 Certified

SYLLABUS

B.Sc. Computer Science
S. Y. B. Sc. Computer Science

As per National Education Policy 2020
Academic Year 2026-27

Mahatma Education Society's College Code: 870
PILLAI HOC COLLEGE OF ARTS, SCIENCE & COMMERCE
Pillai HOCL Educational Campus, HOC Colony, Rasayani, Via. Panvel, Dist. Raigad. Pin 410207
Tel: 02192 - 669000 / 01 / 02 / 03 / 04 / 05 / 06 / 07 / 08 / 09 Website : www.phcasc.ac.in Email : phcasc@mes.ac.in
(NAAC Accredited 'A+' Grade , CGPA – 3.26 in Cycle 2 & ISO 9001:2015 Certified)
Affiliated to the University of Mumbai, Approved by Government of Maharashtra
(AUTONOMOUS COLLEGE)

Sr.No.	Name	Designation	Signature
1	Dr. Swapna Kadam	Vice Chancellor Nominee	
2	Dr. Annie Rajan	Subject Expert	
3	Dr. Homraj Patelpaik	Subject Expert	
4	Mr. Swapnil H. Patil	Industry Representative	
5	Mr. Akash Ghadge	Alumni Representative	
6	Dr. Rinkoo Shantnu	Principal	
7	Ms. Priyanka Sorte	Chairperson	
8	Ms.Jyoti Borade	Member	
9	Ms. Sonali Dagwar	Member	
10	Ms. Rutuja Kondalkar	Member	
11	Mr. Yash Karkhanis	Member	
12	Ms. Aishwarya Mokal	Member	
13	Ms. Kranti Vartak	Member	

Introduction

A **B.Sc. in Computer Science** is a three-year undergraduate program designed to bridge the gap between theoretical knowledge and practical application, with a strong focus on developing a deep understanding of computer systems and networks. The comprehensive curriculum covers a wide range of software and hardware technologies and their real-world applications, enabling students to build strong problem-solving and program design skills while enhancing analytical thinking. Throughout the course, students gain technical proficiency by designing and developing computer programs in diverse and emerging areas such as web development, security, cloud computing, and data science, while applying core computer science principles and software engineering practices. The program also emphasizes innovation and professionalism by encouraging students to stay updated with industry trends, engage in lifelong learning, work effectively in teams, manage projects, and uphold ethical standards in the use of cyber systems. This interdisciplinary training prepares graduates for technical employment, higher education, entrepreneurship, and new career paths, supported by strong placement opportunities, incubation assistance, and exposure to modern advancements and emerging sub-fields in computer science.

Programme Outcomes (POs)

PO. No.	PO Title	POs in brief
PO1	Fundamental Knowledge Acquisition	Graduates will demonstrate a comprehensive and foundational knowledge of their chosen discipline along with an awareness of interdisciplinary connections.
PO2	Critical Thinking and Analytical Reasoning	Graduates will be able to analyse complex problems, synthesize data from multiple sources (qualitative and quantitative), and employ logical reasoning to formulate well-supported conclusions and arguments.
PO3	Effective Communication	Graduates will exhibit proficiency in both written and oral communication, articulating ideas clearly, persuasively, and ethically to diverse audiences
PO4	Problem Solving	Graduates will possess the ability to identify, formulate, and design solutions for real-world problems in their professional or social contexts, applying relevant theoretical knowledge and practical skills.
PO5	Information and Digital Literacy	Graduates will demonstrate the capability to locate, evaluate, and effectively use information from various sources, and utilize modern tools and Information and Communication Technology (ICT) for professional and academic tasks.
PO6	Research Skills and Scientific Temperament	Graduates will develop a sense of inquiry and research methodology, including the ability to design experiments (where applicable), collect and analyse data, and interpret results while maintaining scientific rigor and intellectual honesty.
PO7	Ethical Reasoning and Professional Integrity	Graduates will recognize ethical dilemmas, commit to professional and academic ethics, and demonstrate an understanding of moral and social responsibilities in their personal and professional conduct.
PO8	Employability and Professional Skills	Graduates will acquire the necessary job-ready skills, managerial competencies, and professional values to secure gainful employment or pursue advanced education in their respective fields.
PO9	Environmental and Sustainability Consciousness	Graduates will understand the importance of environmental conservation and sustainable development, displaying responsibility toward ecological challenges and advocating for healthy environmental practices.
PO10	Life-Long Learning	Graduates will develop the capacity for independent and self-directed learning to continuously upgrade their knowledge and skills, enabling them to adapt to rapid technological and societal changes.

PO11	Civic and Social Responsibility	Graduates will act as responsible citizens with an informed awareness of constitutional values, engaging proactively in community development and addressing social needs.
PO12	Empathy and Social Intelligence	Graduates will be able to cultivate and demonstrate affective, interpersonal, social and emotional intelligence.

Programme Specific Outcomes (PSOs)

PSOs. No.	PSO Title	PSOs in brief
PSO1	Technical Design & System Development	Graduates will be able to design, develop, and implement reliable computer applications and systems across diverse domains such as networking, web technologies, security, cloud computing, IoT, and data science.
PSO2	Application of Computing Principles & Engineering Practices	Graduates will apply core computer science theories, software engineering principles, tools, and technologies to analyze, model, and solve real-world computing problems efficiently.
PSO3	Innovation, Research Mindset & Professional Ethics	Graduates will demonstrate awareness of current technological trends, foster innovation in problem-solving, and uphold ethical standards in computing, Internet usage, and cyber practices.
PSO4	Teamwork, Project Management & Lifelong Career Growth	Graduates will work effectively as individuals and team members, manage software projects efficiently, and pursue continuous learning to excel in higher studies and evolving IT career opportunities.

Evaluation Pattern

Marking Code	Marking Scheme
A	50 Marks Semester End Exam, 50 Marks Continuous Assessment (distributed within 15 Marks Class Test, 15 Marks Presentation & Assignment, 10 Marks Online Quiz, 10 Marks Attendance & Class Participation)
B	50 Marks Semester End Exam
C	100 marks Continuous Assessment (distributed within 30 Marks Class Test, 30 Marks Presentation & Assignment, 30 Marks Online Quiz, 10 Attendance & Class Participation)
D	50 Marks of Continuous Assessment (distributed within 15 Marks Class Test, 15 Marks Presentation & Assignment, 10 Marks Online Quiz, 10 Marks Attendance & Class Participation)
E	50 Marks Practical Examination (distributed within 30 Marks Practical Module 1 & 2, 10 Marks Journal, 10 Marks Viva)

Course Structure

Semester III							
Course Code	**Course Type**	**Course Title**	**Theory/ Practical**	**Marks**	**Credits**	**Lectures/ Week**	**Evaluation Pattern**
HUSCS208	Major	ABSTRACT DATATYPES AND ALGORITHM ANALYSIS	Theory	100	2	2	A
HUSCS208P	Major - Practical	PRACTICAL(HUSCS208)	Practical	50	1	2	E
HUSCS209	Major	PRINCIPLES OF OPERATING SYSTEM	Theory	100	2	2	A
HUSCS209P	Major - Practical	PRACTICAL(HUSCS209)	Practical	50	2	2	E
HUSCS210	Major	PYTHON BEYOND BASICS	Theory	100	2	2	A
HUSCS210P	Major - Practical	PRACTICAL(HUSCS210)	Practical	50	1	2	E
HUSCS211	Minor	R PROGRAMMING FOR DATA SCIENCE	Theory	100	2	2	A
HUSCS211P	Minor - Practical	PRACTICAL(HUSCS211)	Practical	50	1	2	E
HUSCS212	SEC	MERN STACK DEVELOPMENT	Theory	100	2	2	A
HUSCS212P	SEC - Practical	PRACTICAL(HUSCS212)	Practical	50	1	2	E
	AEC	हऽिंदी भाषा एवि साहऽत्य सिंवर्धन	Theory	50	2	2	D
	OE	RESEARCH METHODOLOGY	Theory	100	3	3	C
	CC	EXTENSION / NSS	-	50	2	2	D
Total				**950**	**22**		**

Abbreviations:

SEC: Skill Enhancement Course
AEC: Ability Enhancement Course
VAC: Value Added Course
VEC: Value Education Course
OE: Open Elective
CEP: Community Engagement Project

BOS	**Mathematics, Statistics and Computer Application**				
Course	**Abstract Datatypes and Algorithm Analysis**				
Course Code	**HUSCS208**	**Level**	**5.0**		
		Type	**Theory**	**Practical**	**Total**
Semester	**III**	**Credits**	2	1	3
Type	**Major**	**No of Teaching Hours**	30	30	60
Evaluation Pattern	**Total Marks**	**Semester End**	**Continuous**	**Practical**	
	150	50	50	50	

Learning Objectives	
1	To introduce fundamental concepts of data structures and abstract data types
2	To develop understanding of linear and non-linear data structures
3	To enable analysis of time and space complexity using Big-O notation
4	To provide practical knowledge of commonly used data structures in computing
5	To apply data structures to solve real-world computational problems

Course Outcomes	
After successful completion of this course, students would be able to: -	
CO1	Understand and create Abstract Data Types (ADT) and analyze algorithms using complexity and Big-O notation.
CO2	Implement linear data structures and solve problems like expression conversion and job scheduling.
CO3	Implement advanced non-linear data structures such as AVL Trees, Heaps, and Graphs, and apply algorithms like Dijkstra's Algorithm.
CO4	Apply hashing techniques and priority queues to solve real-world computational problems efficiently.

Modules at Glance

Module No.	Content	No. of Hours	CO Mapping
1	Linear Data Structures	15	CO1, CO2, CO3
2	Non-Linear Data Structures	15	CO4, CO5

Syllabus

Module No.	Content	No. of Lectures
1	**Abstract Data Type:** Different Data Types, different types of data structures & their classifications, Introduction to ADT, Creating user-specific ADT, Complexity, Big O notation, Algorithm Analysis. **Stacks:** Stack ADT for Stack, Introduction to Stack, Advantages & Disadvantages, Applications of stack like balanced delimiter, prefix to postfix notation **Queues:** Queue ADT, Introduction to Queue, Advantages & Disadvantages, linked representations. Circular Queue operations, Priority Queues, Dequeues, applications of queue like job scheduling queues **Linked Structures:** ADT for linked list, Advantages & Disadvantages, Singly Linked List-Traversing, Searching, Prepending and Removing Nodes, Doubly linked list, Insertion and deletion of nodes at various positions	15
2	**Trees:** ADT for Tree Structure, Introduction to Trees, Advantages & disadvantages, Binary Tree-Properties, Implementation and Traversals, Binary Search Tree, AVL Trees, Applications of Tree like Huffman Coding, **Priority Queues & Heaps:** Priority Queue, Priority Queue ADT, Advantages and Disadvantages, Applications, Heaps, types of heaps, Heapifying the element, **Graph:** Introduction, Graph ADT, Graph Representation using adjacency matrix and adjacency list, Graph operations like insertion and deletion of nodes, Graph Traversals using BFS & DFS, Dijkstra's Algorithm. **Hashing:** Hash Table ADT,Concept of hashing, hash table, hash functions, collision, collision avoidance techniques, Applications of hashing	15
Case Study Scenario		
M1	**Scenario:** A computer lab prints documents for many students using a single printer. Print requests arrive at different times and are processed in the order they are received. Each request is placed in a waiting list, and the printer always handles the first request in the list before moving to the next one. **Question:** Identify the appropriate data structure to manage the print requests and explain why it is suitable. Describe how insertion and deletion operations would work in this scenario.	
M2	**Scenario:** A delivery company needs to find the shortest route between its warehouse and multiple delivery locations in a city. The city map consists of intersections connected by roads, each having a different travel distance. The company wants to minimize fuel consumption and delivery time. **Question:** Which data structure and algorithm should be used to solve this problem? Explain how the method finds the shortest path from the warehouse to all delivery points	

Reference Books:

1. Ellis Horowitz, Sartaj Sahni, Susan Anderson-Freed -*Fundamentals of Data Structures in C*
2. Mark Allen Weiss — *Data Structures and Algorithm Analysis*
3. Seymour Lipschutz — *Data Structures (Schaum's Outline Series)*
4. Reema Thareja — *Data Structures Using C*
5. Goodrich, Tamassia, Goldwasser — *Data Structures and Algorithms in C++*

Semester End Evaluation (50 Marks) **Time : 2 Hours**

Paper Pattern

Question No.	Questions	Total Marks : 50
Q1	Attempt 3 out of 5	15
Q2	Attempt 3 out of 5	15
Q3	Attempt 3 out of 5	15
Q4	Case Study	05

Practical Syllabus

Sr. No	List of Practical	No. of. Lectures	CO Mapping
1	**Array Operations** Write a program to perform insertion, deletion, and linear search on an array.	3	CO1, CO3, CO5
2	**Singly Linked List Operations** Implement creation, traversal, insertion, deletion, and searching in a singly linked list.	3	CO1, CO3, CO5
3	**Doubly Linked List Operations** Implement insertion and deletion at beginning, end, and specific position in a doubly linked list.	3	CO1, CO3, CO5
4	**Stack Implementation Using Array** Implement stack operations (push, pop, peek) using array representation.	3	CO3,CO5
5	**Applications of Stack** Write a program to check balanced parentheses and/or evaluate postfix expression.	3	CO3, CO5
6	**Queue Implementation Using Array** Implement linear queue and circular queue operations (enqueue, dequeue, display).	3	CO3,CO5
7	**Binary Tree Traversals** Construct a binary tree and perform preorder, inorder, and postorder traversals.	3	CO4, CO5
8	**Binary Search Tree (BST) Operations** Implement insertion, deletion, searching, and traversal in a BST.	3	CO4, CO5
9	**Heap and Priority Queue** Implement a max heap or min heap and perform heapify, insertion, and deletion operations.	3	CO4, CO5
10	**Graph Traversal and Shortest Path** Implement BFS and DFS traversals of a graph .	3	CO2, CO4, CO5

Semester End Practical Evaluation **Time: 2 Hours**

Question No.	Questions	Total Marks
Q.1	Program	30
Q.2	Journal	10
Q.3	Viva & Attendance	10

BOS	**Mathematics, Statistics and Computer Application**				
Course	**Principles of Operating System**				
Course Code	**HUSCS209**	**Level**	**5.0**		
		Type	**Theory**	**Practical**	**Total**
Semester	**III**	**Credits**	2	1	3
Type	**Major**	**No of Teaching Hours**	30	30	60
Evaluation Pattern	**Total Marks**	**Semester End**	**Continuous**	**Practical**	
	150	50	50	50	

Learning Objectives	
1	To understand the fundamentals and structure of modern operating systems.
2	To analyze process management and CPU scheduling techniques.
3	To understand multithreading and concurrency control mechanisms.
4	To evaluate memory management and virtual memory techniques.
5	To understand deadlock handling and system reliability.
6	To familiarize students with modern file systems used in industry.

Course Outcomes	
After successful completion of this course, students would be able to: -	
CO1	Understand the fundamentals of operating systems, including architecture, system calls, and protection mechanisms.
CO2	Analyze process and thread management, including CPU scheduling algorithms and multithreading.
CO3	Apply concurrency and synchronization techniques to manage race conditions and process coordination.
CO4	Develop memory management, deadlock handling, and file system concepts used in modern operating systems.

Modules at Glance

Module No.	**Content**	**No. of Hours**	**CO Mapping**
1	Modern Operating Systems & Process Management	15	CO1,CO2
2	Memory, Reliability & Storage Systems	15	CO3,CO4

Syllabus

Module No.	Content	No. of Lectures
1	**Operating System Fundamentals:** Role and functions of operating systems, types of operating systems (including modern systems), operating system services and utilities, user mode and kernel mode, system calls, basics of protection and security. **Process & Thread Management:** Process concept, process states, process control block, Inter-process Communication, scheduling algorithms (FCFS, SJF, Priority, Round Robin),Threads and multithreading in modern operating systems. **Concurrency & Synchronization:** Concurrency in operating systems, Race conditions and critical section problem, semaphores and mutex locks, classical synchronization problems.	15
2	**Memory Management:** Main memory organization, swapping, paging, Contiguous Memory Allocation, Structure of page table, virtual memory, demand paging, page replacement algorithms (FIFO, LRU, Optimal, Clock), frame allocation, thrashing. **Deadlock Handling:** Deadlock concept, Deadlock conditions, deadlock prevention, deadlock avoidance (Banker's algorithm),deadlock detection & recovery from deadlock, deadlocks in real-world systems. **File Systems & Storage Management:** File concept, file attributes, file access methods, directory structures, file system architecture, disk allocation methods, overview of modern file systems (EXT4, NTFS, FAT).	15
Case Study Scenario		
M1	An operating system must handle multiple user processes where response time is critical and all processes should get equal CPU access. Which CPU scheduling algorithm should be used and why	
M2	A system uses demand paging with FIFO page replacement. Given a reference string and number of frames, calculate the number of page faults.	

Reference Books:

1. Silberschatz, A., Galvin, P. B., & Gagne, G. (2018). *Operating system concepts* (10th ed.). Wiley.
2. Tanenbaum, A. S., & Bos, H. (2015). *Modern operating systems* (4th ed.). Pearson Education.
3. Stallings, W. (2018). *Operating systems: Internals and design principles* (9th ed.). Pearson Education.
4. Dhamdhere, D. M. (2007). *Operating systems: A concept-based approach* (2nd ed.). McGraw-Hill Education.
5. Arpaci-Dusseau, R. H., & Arpaci-Dusseau, A. C. (2018). *Operating systems: Three easy pieces*. Arpaci-Dusseau Books.

Semester End Evaluation (50 Marks) **Time: 2 Hours**

Paper Pattern

Question No.	Questions	Total Marks: 50
Q1	Attempt 3 out of 5	15
Q2	Attempt 3 out of 5	15
Q3	Attempt 3 out of 5	15
Q4	Case Study	05

Practical Syllabus

Sr. No	List of Practical	No. of. Lectures	CO Mapping
1	Write a program to demonstrate process creation using fork() system call and illustrate parent–child process execution.	3	CO1
2	Write a program to implement Inter-Process Communication using pipes between two processes.	3	CO1
3	Write a program to demonstrate multithreading using POSIX threads (pthreads) for concurrent execution of tasks.	3	CO 2
4	Write a multithreaded program to generate the Fibonacci sequence using thread libraries.	3	CO2
5	Write a program to implement the Bounded Buffer (Producer–Consumer) problem using semaphores or mutex locks.	3	CO3
6	Write a program to simulate the Readers–Writers synchronization problem.	3	CO3
7	Write a program to implement CPU scheduling algorithms (FCFS and Round Robin) and calculate waiting time and turnaround time.	3	CO2
8	Write a program to implement Banker's Algorithm for deadlock avoidance and determine safe sequence.	3	CO4
9	Write a program to simulate Page Replacement Algorithms (FIFO and LRU) and analyze page faults.	3	CO4
10	Write a program demonstrating basic file system operations such as file creation, reading, writing, and deletion using system calls.	3	CO4

Semester End Practical Evaluation **Time: 2 Hours**

Question No.	Questions	Total Marks
Q.1	Program	30
Q.2	Journal	10
Q.3	Viva & Attendance	10

BOS	**Mathematics, Statistics and Computer Application**				
Course	**Python Beyond Basics**				
Course Code	**HUSCS210**	**Level**	**5.0**		
		Type	**Theory**	**Practical**	**Total**
Semester	**III**	**Credits**	2	1	3
Type	**Major**	**No of Teaching Hours**	30	30	60

Evaluation Pattern	**Total Marks**	**Semester End**	**Continuous**	**Practical**
	150	50	50	50

Learning Objectives	
1	Understand and apply core Object-Oriented Programming concepts including classes, objects, constructors, inheritance, and polymorphism.
2	Design and implement database-driven and GUI-based applications using Python.
3	Develop multithreaded and network-based Python applications using synchronization and REST API concepts.
4	Analyze, process, and visualize data using Python Data Science libraries such as NumPy, Pandas, Matplotlib, and SciPy.

Course Outcomes	
After successful completion of this course, students would be able to: -	
CO1	Apply object-oriented programming concepts by implementing constructors, destructors, and different types of methods in Python programs.
CO2	Build database-driven and GUI-based Python applications and perform CRUD operations effectively.
CO3	Develop multithreaded and networking applications using TCP/IP and integrate REST APIs.
CO4	Analyze data using data science tools and visualize results while performing basic scientific computing tasks.

Modules at Glance

Module No.	**Content**	**No. of Hours**	**CO Mapping**
1	Advanced Object-Oriented Programming and Multithreading in Python	15	CO1,CO2
2	Advanced Python Application Development	15	CO3,CO4

Syllabus

Module No.	Content	No. of Lectures
1	**OOPs Fundamentals:** Introduction to Object-Oriented Programming, Limitations of Procedure-Oriented Programming, Core features of OOPs (Encapsulation, Abstraction, Inheritance, Polymorphism), Creating Classes and Objects, Types of variables (Instance & Class), Types of methods (Instance, Class, Static) **Constructors and Destructors:** Constructors, parameterized constructors, and destructors. **Graphical User Interface (GUI) Development:** Creating GUI applications in Python, widget classes, and layout managers.	15
2	**Database Programming with Python:** Basics of DBMS, CRUD operations (Create, Read, Update, Delete), and creating and managing database tables using Python. **Networking with Python:** TCP/IP protocol basics, REST APIs, and web page operations using Python. **Data Science Tools with Python:** Introduction to NumPy, data analysis using Pandas, data visualization with Matplotlib, and scientific computing using SciPy	15
Case Study Scenario		
M1	Smart Hospital Management System (SHMS) A hospital wants to develop a Smart Hospital Management System that allows patients to register online, book appointments, access medical reports, and pay bills securely through a GUI application connected to a centralized database server. Doctors and administrators can manage schedules, update prescriptions, and monitor patient records in real time. Objectives Automate patient registration and appointment booking Maintain digital medical records Manage billing and payments Provide secure access to doctors and patients	
M2	Smart E-Learning Management System (SELMS) An educational institution wants to develop a Smart E-Learning Management System that allows students to enroll in courses, access study materials, submit assignments, attend live classes, and track performance through a GUI/web-based application connected to a database server. Objectives Manage student registration and course enrollment Provide digital learning resources Conduct online assessments	

Reference Books:

1. Practical Programming: An Introduction to Computer Science Using Python 3, Paul Gries , Jennifer Campbell, Jason Montojo, Pragmatic Bookshelf, 2nd Edition, 2014
2. Programming through Python, M. T Savaliya, R. K. Maurya& G M Magar, Sybgen Learning India, 2020
3. Python: The Complete Reference, Martin C. Brown, McGraw Hill, 2018
4. Beginning Python: From Novice to Professional, Magnus Lie Hetland, Apress, 2017
5. Programming in Python 3, Mark Summerfield, Pearson Education, 2nd Ed,2018

Semester End Evaluation (50 Marks) **Time: 2 Hours**

Paper Pattern

Question No.	Questions	Total Marks: 50
Q1	Attempt 3 out of 5	15
Q2	Attempt 3 out of 5	15
Q3	Attempt 3 out of 5	15
Q4	Case Study	05

Practical Syllabus

Sr. No	List of Practical	No. of. Lectures	CO Mapping
1	Write a Python program to create a class with instance variables, class variables, instance methods, class methods, and static methods.	3	**CO1**
2	Develop a program demonstrating default constructor, parameterized constructor and destructor.	3	**CO1**
3	Create a base class and derived class to demonstrate inheritance and method overriding.	3	**CO1**
4	Design a simple student registration form using Tkinter with labels, textboxes, and buttons.	3	**CO2**
5	Creating a Student Registration Form Using Tkinter	3	**CO2**
6	Create a database and perform Create, Read, Update, Delete operations using SQLite in Python.	3	**CO2**
7	Develop a basic TCP server and client program using Python socket programming.	3	**CO3**
8	Fetch data from a public REST API using requests library and display the response in Python.	3	**CO3**
9	Load a CSV file using Pandas and perform basic analysis (mean, sum, filtering).	3	**CO4**
10	Plot bar chart and line chart using Matplotlib for a given dataset.	3	**CO4**

Semester End Practical Evaluation **Time: 2 Hours**

Question No.	Questions	Total Marks
Q.1	Program	30
Q.2	Journal	10
Q.3	Viva & Attendance	10

BOS	**Mathematics, Statistics & Computer Application**				
Course	**R Programming for Data Science**				
Course Code	**HUSCS211**	**Level**	5.0		
			Theory	**Practical**	**Total**
Semester	III	**Credit**	02	01	03
Type	Minor	**No of Teaching hours**	**30**	**30**	**60**

Evaluation/ Assessment	**Total Marks**	**Semester End**	**Continuous Evaluation**	**Practical Examination**
	150	**50**	**50**	**50**

Learning Objectives	
1	To provide students with a strong foundation in R programming basics, including its user interface, objects, functions, and scripts.
2	To help students master the use of R packages and help pages to carry out data analysis tasks more efficiently.
3	To enable students to work with R objects such as atomic vectors, matrices, arrays, lists, data frames, and understand loading and saving data for effective analysis, with a focus on efficient data manipulation using the dplyr package.
4	To teach students how to create informative graphs and plots using the ggplot package.

Course Outcomes	
CO1	Students will have the ability to write their own functions, manage arguments and understand scripts in the R programming language.
CO2	Students will proficiently use packages and help pages in R programming for the better resolution of the challenges of data analysis
CO3	Students will gain a strong foundation in R objects, different data frames, working with loading data, saving data and will be able to efficiently manipulate data frames using dplyr.
CO4	Students will be able to work efficiently on informative and visually appealing plots using the ggplot2

Modules At Glance

Module No.	Content	No. of Hours	Mapping with CO
1	Basics Of R Programming	15	1,2
2	Working with R	15	3,4,5
		30	

Syllabus

Module No.	Content	No. of Hours
1	1. R Basics: The R User interface, objects, functions, sample with replacement writing own functions arguments scripts 2. Packages and Help pages: Packages, Help packages 3. R Objects: Atomic Vectors, Attributes, Matrices, Arrays, Class, Coercion, Lists, data frames, loading data, saving data 4. Modifying Values: Changing values in place, logical subsetting, missing information 5. Environments: Introduction, working with environments, scoping rules, assignment, evaluation, closures	15
2	1. Programs: Strategy, if statements, else statements, lookup tables, code comments 2. Loops: Expected values, expand.grid, for loops, while loops, repeat loops, 3. Working with Strings – string: Counting String Patterns, Splitting Strings, Capitalizing Strings, Wrapping, Padding, and Trimming, Detecting Substrings, Extracting Substrings, Transforming Strings. 4. R for data science: Data Manipulation with dplyr(Filtering, selecting, mutating, arranging, summarizing, grouping data frames) 5. Data Visualization with ggplot2: Introduction to ggplot2, creating scatter plots, line plots, bar charts, histograms, box plots, customizing plots.	15
Case Study Scenarios		
M1	**Academic Performance and Behavioral Analysis** **Scenario:**A university wants to understand the relationship between student attendance, assignment submission patterns, and final exam scores across different departments.	
M2	**Retail Sales and Regional Trend Analysis** **Scenario:** A retail chain needs to analyze its sales performance across different regions (North, South, East, West) and product categories to prepare for an end-of-year board meeting.	

Text and References:

1. Hands-On Programming with R, Garrett Grolemund, O'Reilly, 2014
2. R Programming: A Step-by-Step Guide for Absolute Beginners. Daniel Bell, Guzzler Media, 2020
3. R in Action (3rd Edition) by Robert Kabacoff.
4. The Art of R Programming by Norman Matloff
5. Advanced R (2nd Edition) by Hadley Wickham.

Semester End Evaluation (50 Marks)

Paper Pattern **Time: 2 Hr**

Question No	Questions	Total Marks: 50
Q1	Attempt any 3 out of 5	**15**
Q2	Attempt any 3 out of 5	**15**
Q3	Attempt any 3 out of 5	**15**
Q4	Case Study	**05**

Syllabus

	List of Practical	No. of. Lectures	CO Mapping
1	**Practical 1: R Basics and Data Types** a) Write R code to create a vector of student names and another vector of their corresponding scores in a class test. Combine these into a data frame and display the first 5 rows. b) Create a list containing a student's name, ID, and a vector of their marks in three different subjects. Access and print the student's name and their marks in the second subject.	CO3	**3**
2	**R Functions** a) Write an R function to calculate the factorial of a given number. Demonstrate its use by calculating the factorial of 5 and 7. b) Explore the use of default argument values in R functions. Create a function with some arguments having default values and demonstrate how to call the function with and without specifying those arguments.	CO1	**3**
3	**Working with Matrices and Arrays** a) Create two matrices, A and B, of size 3x3 with random numbers. Perform matrix addition, subtraction, and multiplication. Calculate the transpose of matrix A. b) Create a 3x3x3 array and demonstrate how to access specific elements and slices of the array.	CO3	**3**
4	**Control Structures - If/Else and Loops** a) Write an R program to determine if a number is prime or not. b) Use a for loop to iterate through a vector of numbers and print whether each number is even or odd.	CO1	**3**
5	**Working with Data Frames** a) Load a CSV file into an R data frame. Display the structure of the data frame (number of rows, columns, data types). b) From the loaded data frame, extract all rows where a specific column (e.g., "Salary") is above a certain threshold (e.g., 50000) and save the result to a new CSV file.	CO3	**3**
6	**Working with dplyr**	CO3	**3**

	Using the dplyr package, perform the following operations on the student data frame: 1. Filter the data to include only students from the "Computer Science" department. 2. Select the Name, Department, and MathScore columns. 3. Create a new column called AverageScore that calculates the average of MathScore, ScienceScore, and EnglishScore for each student. 4. Arrange the data frame in descending order of AverageScore. 5. Group the data by Department and calculate the average MathScore for each department.		
7	**List Manipulation** a) Create a nested list representing a hierarchical structure (e.g., a company organization). Access and modify elements at different levels of the list. b) Write a function that takes a list of numeric vectors as input and returns a new list containing the mean of each vector.	CO3	**3**
8	**R for Data Science Basics** a) Given a vector of numerical data, calculate the mean, median, and standard deviation using R functions. b) Create a simple scatter plot in R using two numerical vectors of the same length. Label the axes appropriately.	CO2, CO4	**3**
9	**Text Manipulation** Performing Text Manipulation using str_sub(),str_trim() & str_to_lower(),str_detect()str_replace_all()str_extract()str_pad()str_split_fixed( )str_c() or str_glue() and str_split() in R	CO2	**3**
10	**Data Aggregation** a) Using a sales data frame with columns like "Region", "Product", and "Sales", calculate the total sales for each region. b) From the same sales data frame, calculate the average sales for each product within each region.	CO3	**3**

Practical Evaluation (50 Marks)

	Assessment/ Evaluation	Marks
1	Practical (Module 1 & Module 2)	30
2	Journal	10
3	Viva & Attendance	10

BOS	**Mathematics, Statistics and Computer Application**				
Course	**MERN Stack Development**				
Course Code	**HUSCS212**	**Level**	5.0		
			Theory	**Practical**	**Total**
Semester	III	**Credit**	02	02	04
Type	SEC	**No of Teaching hours**	**30**	**30**	**60**

Evaluation/ Assessment	**Total Marks**	**Semester End**	**Continuous evaluation**	**Practical Examination**
	150	**50**	**50**	**50**

Learning Objectives	
1	To understand the fundamentals of NoSQL databases and develop skills in designing and managing data using MongoDB with Mongoose.
2	To learn backend application development using Express.js including RESTful API design, middleware usage, and error handling.
3	To understand server-side programming concepts using Node.js, including asynchronous programming and request handling.
4	To develop dynamic and interactive frontend applications using React, including component-based architecture, state management, routing, forms, validation, and integration with backend services.

Course Outcomes	
CO1	Design and implement NoSQL databases by creating collections, schemas, and performing CRUD operations using MongoDB and Mongoose.
CO2	Develop backend applications and RESTful APIs using Express.js, incorporating middleware, routing, and error handling techniques.
CO3	Build server-side applications using Node.js by applying asynchronous programming, modularization, and HTTP server concepts.
CO4	Create responsive and interactive frontend applications using React and integrate them with backend APIs to develop complete full-stack applications.

Modules At Glance

Module No.	Content	No. of Hours	Mapping with CO
1	MongoDB & Mongoose (Introduction to NoSQL, installation using MongoDB Compass, collections, documents, CRUD operations, schema design, validation, indexing), Express.js Framework (Introduction to Express.js, middleware, RESTful APIs, routing, error handling, environment configuration)	15	CO1, CO2
2	React Framework (Introduction to React, components, JSX, state, props, hooks, routing, forms, validation, HTTP integration, CRUD), Node.js Fundamentals (Introduction to Node.js, npm, modules, event loop, HTTP server, routing)	15	CO3, CO4
		30	

Syllabus

Module No.	Content	No. of Hours
1	**MongoDB & Mongoose (Database Layer)** Introduction to NoSQL and MongoDB, installation and setup using MongoDB Compass, understanding collections and documents, performing CRUD operations using Mongo Shell and Compass, introduction to Mongoose, defining schemas and models, data validation techniques, indexing basics **Express.js Framework (Application Layer)** Introduction to Express.js, creating Express applications, understanding middleware concepts including built-in, custom and third-party middleware, RESTful API design using GET, POST, PUT and DELETE methods, routing and use of router modules, error handling mechanisms, environment configuration using dot net.	15

2	**<u>React Framework (Frontend Development)</u>** Introduction to React and React CLI tools, understanding React project structure, components, JSX, and styling, one-way data binding, conditional rendering and list rendering, state and props management, functional components and hooks (useState, useEffect), routing using React Router, form handling and validation, HTTP requests using fetch/axios for API integration, handling asynchronous data, basic CRUD operations with backend integration **<u>Node.js Fundamentals (Server Runtime)</u>** Introduction and installation of Node.js, understanding npm and package.json, working with node modules and module exports, event loop and asynchronous programming concepts, creating basic HTTP server, handling requests and routing, introduction to backend integration concepts	15
M1	A college library wants to develop a backend system to manage book records, members, and issue/return transactions. The system should allow the librarian to add new books, update book details, delete records of lost or outdated books, and view all available books. It should also manage member details and track which books are issued and returned. The backend of the system is to be developed using Express.js, while data will be stored in MongoDB using Mongoose for schema design and validation. Using Module 1 concepts, explain how you would design and implement the system including creating collections for books, members, and transactions, defining schemas and models using Mongoose, implementing CRUD operations through RESTful APIs (GET, POST, PUT, DELETE), using middleware for request processing and validation, handling errors effectively in Express, organizing routes using router modules, and configuring environment variables using dotenv for database connection and server settings.	
M2	A college wants to develop an Online Feedback System where students can submit feedback for courses and faculty members. The system should provide a user-friendly interface for submitting feedback and allow administrators to view and analyze responses. The frontend will be developed using Angular, and the server-side runtime will use Node.js to handle requests and communicate with backend APIs. Using Module 2 concepts, describe how you would design and implement the system including creating Angular components for feedback forms and dashboards, setting up Angular routing for navigation between pages, implementing template-driven or reactive forms with proper validation, using services and HttpClient to send and retrieve data from backend APIs, handling asynchronous operations using observables and async pipe, creating a basic Node.js server to handle requests, managing API integration, handling errors in HTTP communication, and organizing the application for scalability and maintainability.	

Books and References:

1. Web Development with MongoDB and Node.js by Mike Wilson, published by O'Reilly Media, 2nd Edition, 2020.

2. Express in Action by Evan Hahn, published by Manning Publications, 1st Edition, 2016.

3. Learning React by Alex Banks and Eve Porcello, published by O'Reilly Media, latest edition.

4. MongoDB: The Definitive Guide by Shannon Bradshaw, Eoin Brazil and Kristina Chodorow, published by O'Reilly Media, 3rd Edition, 2019.

5. Pro Git by Scott Chacon and Ben Straub, published by Apress, 2nd Edition, 2014.

Semester End Evaluation (50 Marks) **Time : 2 Hours**

Paper Pattern

Question No.	Questions	Total Marks : 50
Q1	Attempt 3 out of 5	15
Q2	Attempt 3 out of 5	15
Q3	Attempt 3 out of 5	15
Q4	Case Study	05

Practical Syllabus

List of Practical		CO Mapping	Hours
1	Write a program to connect to MongoDB using Mongoose and create a schema and model for a collection (e.g., Student/Book)	CO1	3 HRS
2	Develop a program to perform CRUD operations (Create, Read, Update, Delete) using Mongoose	CO1	3 HRS
3	Implement data validation and indexing in MongoDB using schema validation rules in Mongoose	CO1	3 HRS
4	Write a program to create a basic HTTP server using Node.js and handle different routes (/, /about, /contact)	CO3	3 HRS
5	Develop a program demonstrating asynchronous programming in Node.js using callbacks, Promises, and async/await	CO3	3 HRS
6	Create a basic application using Express.js with routing for multiple endpoints and JSON responses	CO2	3 HRS
7	Design and implement RESTful API endpoints (GET, POST, PUT, DELETE) using Express.js integrated with MongoDB	CO2	3 HRS
8	Implement middleware (custom logger/authentication) and error handling in an Express.js application	CO2	3 HRS
9	Develop a React Application using Components, JSX, State and Props	CO4	3 HRS
10	React Application with Routing and Dynamic Data Rendering	CO4	3 HRS

Semester End Practical Evaluation **Time: 2 Hours**

Question No.	Questions	Total Marks
Q.1	Program	30
Q.2	Journal	10
Q.3	Viva & Attendance	10