final

Since 1989

Devoted to Serve

Royal College of Arts, Science and Commerce
(Autonomous)
Affiliated to University of Mumbai

Program: Bachelor of Science
Course: Computer Science
Syllabus for Semester: III and IV

Syllabus for Undergraduate Programme as per National Education Policy (NEP-2020) with effect from Academic year 2025-2026

ROYAL COLLEGE OF ARTS DIST THANE

Principal
ROYAL COLLEGE OF ARTS
SCIENCE & COMMERCE
PENKAR PADA, MIRA ROAD,
DIST : THANE. PIN : 401107.

NEP Credit Structure for Computer Science Department

Level	Sem	Major		Minor	OE	VSC	SEC	AEC	IKS	VEC	OJT/FP /RP/CC /CEP	Cumulative Credits	
		DSC	DSE										
4.5	I	4TH+2PR 6		2TH	2TH+ 2TH	2PR	2PR	2TH	2TH	2TH		22	UG Certificate Cumulative Credit:44
	II	4TH+2PR 6		2TH	2TH+ 2TH	2PR	2PR	2TH		2TH	2 CC	22	
Exit Option: Award of UG Certificate in Major with 40-44 Credits and an Additional 4 Credits Core NSQF Course / Internship OR Continue with Major and Minor													
5	III	8TH+2PR 10		2TH+2PR 4	2TH		2PR	2TH			2 CC	22	UG Diploma Cumulative Credit:88
	IV	8TH+2PR 10		2TH+2PR 4	2TH		2PR	2TH			2 CEP	22	
Exit Option: Award of UG Diploma in Major and Minor with 80-88 Credits and an Additional 4 Credits Core NSQF Course /Internship OR Continue with Major and Minor													
5.5	V	8TH+2PR 10	2	2TH+2TH 4		2					2 FP +2 OJT	22	UG Degree Cumulative Credit:132
	VI	8TH+2PR 10	2	2TH+2TH 4		2					2 FP +2 OJT	22	
Total		**52**	**4**	**20**	**12**	**8**	**8**	**8**	**2**	**4**	**14**	**132**	

COLLEGE OF ARTS DIST THANE

Principal
ROYAL COLLEGE OF ARTS
SCIENCE & COMMERCE
PENKAR PADA, MIRA ROAD,
DIST : THANE. PIN : 401107.

Programme Outcomes (POs) for Computer Science

	On completing B.Sc. Computer Science, the student will be able to:
PO1	Develop understanding and knowledge of the fundamental theories, applications, technologies in computer science.
PO2	Understanding and applying the knowledge of networking, web design, security, cloud computing, IoT.
PO3	Understanding of best practices, standards to develop user interactive and abstract applications.
PO4	Demonstrate proficiency in using current techniques, skills, and tools necessary for computing practice.
PO5	Apply concepts, principles and theories relating to computer science to new situations.
PO6	Apply standard software engineering practices and strategies in real-time software project development.
PO7	Apply technical skills to formulate & design solutions to solve real world problems.
PO8	Pursue higher studies of specialization and to take up technical employment.
PO9	Develop Work Ethics , Communication Skills to present their work effectively and coherently.
PO10	Engage in independent and life-long learning in the background of the rapidly changing IT industry.

Programme Specific Outcomes (PSOs) for Computer Science

	On completing B.Sc. Computer Science, the student will be able to:
PSO1	Demonstrate proficiency in different computing paradigms needed for a proper understanding of computer science.
PSO2	Develop and implement software solutions to effectively address real-world problems in business, science, and social contexts.
PSO3	Build up programming, analytical and logical thinking abilities.
PSO4	Design, develop and analyze the use of the Information security tools and technique for providing security.
PSO5	Demonstrate programs in various languages to enhance computer and internet capabilities.
PSO6	Apply the standard software engineering practices and strategies in real time software project development.

Semester – III				
Course Code	**Course Type**	**Course Title**	**Credits**	**Lectures /Week**
RUCSMJ301	**Major I**	**Operating Systems**	**4**	**4**
RUCSMJ302	**Major II**	**Data Structures**	**4**	**4**
RUCSMJP3	**Major Practicals**	**MJP3: Computer Science Practical**	**2**	**4**
RUCSMN303	**Minor I**	**Statistical Techniques**	**2**	**2**
RUCSMNP304	**Minor II**	**IOT Technologies**	**2**	**2**
RUCSSECP305	**SEC**	**Java Programming**	**2**	**2**
	AEC	**Language**	**2**	**2**
	OE	**Open Elective I**	**2**	**2**
	CC	**NSS, NCC, Sports, Cultural, Yoga, Music, Performing Arts, DLLE**	**2**	**2**

[Abbreviation - OE – Open Electives, VSC – Vocation Skill Course, SEC – Skill Enhancement Course, (VSEC), AEC – Ability Enhancement Course, VEC – Value Education Course, IKS – Indian Knowledge System, OJT – on Job Training, FP – Field Project, CEP – Continuing Education Program, CC – Co-Curricular, RP – Research Project]

Semester – IV				
Course Code	**Course Type**	**Course Title**	**Credits**	**Lectures /Week**
RUCSMJ401	**Major I**	**Theory of Computation**	**4**	**4**
RUCSMJ402	**Major II**	**Computer Networks**	**4**	**4**
RUCSMJP4	**Major Practicals**	**MJP4: Computer Science Practical**	**2**	**4**
RUCSMN403	**Minor I**	**Fundamentals of Data Science**	**2**	**2**
RUCSMNP404	**Minor II**	**Advanced Application Development**	**2**	**2**
RUCSECP405	**SEC**	**Android Application Development**	**2**	**2**
	AEC	**Language**	**2**	**2**
	OE	**Open Elective I**	**2**	**2**
	CC	**NSS, NCC, Sports, Cultural, Yoga, Music,Performing Arts, DLLE**	**2**	**2**

[Abbreviation - OE – Open Electives, VSC – Vocation Skill Course, SEC – Skill Enhancement Course, (VSEC), AEC – Ability Enhancement Course, VEC – Value Education Course, IKS – Indian Knowledge System, OJT – on Job Training, FP – Field Project, CEP – Continuing Education Program, CC – Co-Curricular, RP – Research Project]

SEMESTER III

Course/ Paper Title	**Operating System**
Course offered as	**Major I**
Course Code	**RUCSMJ301**
Semester	**III**
No. of Credits	**4**
No. of lecture Hours/week	**4**

Sr No.	**Course Learning Objectives:**
CLO1	Identify and describe fundamental concepts of operating systems, roles, functions, computing environments, and historical development.
CLO2	Explain key operating system components such as processes, threads, CPU scheduling, memory management, and file systems.
CLO3	Demonstrate the ability to utilize operating system services, system calls, and scheduling algorithms to manage processes, memory, and storage efficiently.
CLO4	Compare and evaluate different operating system structures, synchronization techniques, deadlock handling methods, and file-system implementations to optimize system performance.
CLO5	Identify and describe fundamental concepts of operating systems, roles, functions, computing environments, and historical development.

Course Outcome

	On completing the course, the student will be able to:
CO1	Define and recall fundamental concepts of operating systems, including their roles, functions, computing environments, and historical developments.
CO2	Explain and interpret key operating system components such as processes, threads, synchronization mechanisms, CPU scheduling, memory management, and file systems.
CO3	Implement process management, memory allocation techniques, and file system operations using operating system structures and services.
CO4	Compare and solve different CPU scheduling algorithms, synchronization mechanisms, deadlock handling techniques, and file-system implementations to enhance system performance.

Detailed Syllabus

Module	Title with content	No. of lectures
I	**Introduction to Operating system**: Definition of OS, role of OS, Operations and Functions of OS , Computing Environments, History, Computer hardware, different operating systems. **Operating-System Structures**: Operating-System Services, User and Interface, System Calls, Types of System Calls, Operating-System Structure **Threads:** Overview, Multicore Programming, Multithreading Models.	**15**
II	**Processes**: Process Concept, Process Scheduling, Operations on Processes, Inter process Communication. **Process Synchronization**: General structure of a typical process, race condition, The Critical-Section Problem, Peterson's Solution, Synchronization Hardware, Mutex Locks, Semaphores, Classic Problems of Synchronization, Monitors. **CPU Scheduling**: Basic Concepts, Scheduling Criteria, Scheduling Algorithms (FCFS, SJF, SRTF, Priority, RR, Multilevel Queue Scheduling, Multilevel Feedback Queue Scheduling), Thread Scheduling.	**15**
III	**Deadlocks**: System Model, Deadlock Characterization, Methods for Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, Recovery from Deadlock. **Main Memory**: Background, Logical address space, Physical address space, MMU, Swapping, Contiguous Memory Allocation, Segmentation, Paging, Structure of the Page Table. **Virtual Memory:** Background, Demand Paging, Copy-on-Write, Page Replacement, Allocation of Frames, Thrashing.	**15**
IV	**Mass-Storage Structure**: Overview, Disk Structure, Disk Scheduling, Disk Management. **File-System Interface**: File Concept, Access Methods, Directory and Disk Structure, File-System Mounting, File Sharing **File-System Implementation**: File-System Structure, File-System Implementation, Directory Implementation, Allocation Methods, Free-Space Management.	**15**

Textbooks:

1. Abraham Silberschatz, Peter Galvin, Greg Gagne, Operating System Concepts, Wiley, 2021

Additional References:

1. Achyut S. Godbole, Atul Kahate, Operating Systems, Tata McGraw Hill, 2017
2. Naresh Chauhan, Principles of Operating Systems, Oxford Press, 2014
3. Andrew S Tanenbaum, Herbert Bos, Modern Operating Systems, 4e Fourth Edition, Pearson Education, 2016

Course/ Paper Title	**Data Structure**
Course offered as	**Major II**
Course Code	**RUCSMJ302**
Semester	**III**
No. of Credits	**4**
No. of lecture Hours/week	**4**

Sr No.	**Course Learning Objectives:**
CLO1	Understand and Implement core data structures and their basic operations.
CLO2	Describe use of advanced data structures and Priority Queues for complex tasks.
CLO3	Study algorithm design, performance analysis, and apply efficient algorithms for sorting, searching, and graph problems.
CLO4	Understand hashing techniques, collision resolution, and memory representation to optimize data storage and retrieval.

Course Outcome

On completing the course, the student will be able to:	
CO1	Implement and manipulate fundamental data structures and perform related operations effectively.
CO2	Select and apply the most suitable data structures to solve real-world computing problems efficiently.
CO3	Analyze and design efficient algorithms for sorting, searching, and graph traversal, optimizing performance for various applications.
CO4	Understand and apply hashing techniques and memory representation strategies to optimize data storage, retrieval, and collision resolution in practical applications.

Detailed Syllabus

Module	Title with content	No. of lectures
I	**Introduction:** Data and Information, Different Data Types, Data Structure, Classification of Data Structures, Primitive Data Types, Abstract Data Types, Operations on Data Structure, Algorithm. **Array:** Concept of One Dimensional Array and Two Dimensional Array, Traversing, Insertion, Deletion, Merging of Arrays, Memory, General Multi-Dimensional Arrays **Stacks:** Stack ADT for Stack, Advantages & Disadvantages, Applications of stack like balanced delimiter, prefix to postfix notation.	**15**
II	**Queues:** Queue ADT, Advantages & Disadvantages, linked representations, applications of queue like job scheduling. **Priority Queues**: Priority Queue, Priority Queue ADT, Advantages and Disadvantages, Applications. **Linked List**: ADT for linked list, Advantages & Disadvantages, Singly and doubly linked list-Traversing, Searching, Prepending and Removing Nodes.	**15**
III	**Tree**: Tree, Binary Tree and its Properties, Memory Representation, Operations and Reconstruction of Binary Tree from its Traversals, Huffman Algorithm, Binary Search Tree, Operations on Binary Search Tree. **Heap:** Memory Representation and Operation on Heap, Heap Sort. **Advanced Tree Structures**: Red Black Tree, Operations Performed on Red Black Tree, AVL Tree, Operations performed on AVL Tree, 2-3 Tree, B-Tree, Applications of Tree.	**15**
IV	**Graph**: Introduction of Graph, Memory Representation of Graph, Adjacency List or Linked Representation of Graph, Operations Performed on Graph, Graph Traversal, Applications of the Graph, Reachability, Shortest Path Problems, Spanning Trees. **Hashing**: Hash Table ADT, Concept of hashing, hash table, hash functions, Address calculation techniques, Common hashing functions Collision resolution, Probing types, collision avoidance techniques, Applications of hashing.	**15**

Textbooks:

1. Data Structures And Algorithms Made Easy, Narasimha Karumanchi, 2021 Additional
2. Schaum's Outlines Data structure Seymour Lipschutz, Tata McGraw Hill, 2005

References:

1. An Introduction to Data Structure with Applications, Jean – Paul Tremblay and Paul Sorenson Tata MacGraw Hill 2007
2. Data Structures and Algorithms in Python, Michael T. Goodrich, Roberto Tamassia, Michael H. Goldwasser, Wiley, 2016

Course/ Paper Title	**MJP3: Computer Science Practical 3**
Course offered as	**Major Practicals**
Course Code	**RUCSMJP3**
Semester	**III**
No. of Credits	**2**
No. of lecture Hours/week	**4**

Sr No.	Course Learning Objectives:
CLO1	Understand and explain process synchronization techniques to solve concurrency problems.
CLO2	Design and develop multithreaded programs in Java.
CLO3	Implement and Analyze scheduling algorithms to simulate CPU process scheduling.
CLO4	Apply memory management techniques by implementing the Banker's algorithm and page replacement algorithms.
CLO5	Solve concurrency issues by implementing synchronization techniques
CLO6	Demonstrate synchronized access to shared resources and resolve deadlock issues
CLO7	Develop a solid understanding of Abstract Data Types and implement core operations on various data structures.
CLO8	Learn to manipulate Singly Linked Lists, Doubly Linked Lists, and Binary Trees with insertion, deletion, and traversal.
CLO9	Understand and apply concepts of graph theory for building and traversing graphs efficiently.
CLO10	Explore the implementation of basic hash tables and collision handling through chaining.

Course Outcome

	On completing the course, the student will be able to:
CO1	Understand process synchronization and communication techniques.
CO2	Apply multithreading to solve problems like prime number generation and string operations.
CO3	Implement and compare scheduling algorithms.
CO4	Develop and evaluate memory management algorithms
CO5	Demonstrate synchronization to solve concurrency issues.
CO6	Implement and demonstrate understanding of basic data structures.
CO7	Apply tree structures and implement a priority queue.
CO8	Solve problems using graph algorithms.
CO9	Apply advanced algorithms like Huffman coding and hash tables.

CO10	Demonstrate practical programming skills and coding proficiency.

Sr No.	MODULE 1: Operating System Practicals
1	Practical can be implemented either in JAVA or any other programming language. Process Communication: i) Give a solution to the Producer Consumer problem using semaphore. ii) Give a solution to the producer–consumer problem using message passing.
2	i) Write a program to print prime numbers from 1 to 100 using a single thread. ii) Write a program to print even or odd and find the reverse of string using concepts of multithreading.
3	Write a program to demonstrate the concept of synchronized access to shared resources.
4	Give a solution to the readers–writers problem using Java synchronization.
5	Implement FCFS scheduling algorithm in Java.
6	Implement SJF (with no preemption) scheduling algorithm in Java.
7	Implement RR scheduling algorithm in Java.
8	Write a Java program that implements the banker's algorithm.
9	Write a Java program that implements the FIFO page-replacement algorithm.
10	Write a Java program that implements the LRU page-replacement algorithm
Sr No.	**MODULE 2: Data Structure Practicals**
1	Write a program to implement Abstract Data Types (ADT)
2	Write a program to implement Singly Linked list with its operations.
3	Write a program to implement Doubly Linked list with its operations.
4	Write a program to implement Stack with insertion, deletion, traversal operations.
5	Write a program to implement Queue with insertion, deletion, traversal operations.
6	Write a program to implement Priority Queue with its operations.
7	Write a program to implement Binary Tree with its operations.
8	Write a program to implement Huffman Coding.
9	Write a program to implement Graph with its operations.

10	Write a program to implement Travelling Salesman Problem.
11	Write a program to create basic Hash Table for its operations.
12	Write a program to create hash table to handle collisions using overflow chaining

Course/ Paper Title	**Statistical Techniques**
Course offered as	**Minor I**
Course Code	**RUCSMN303**
Semester	**III**
No. of Credits	**2**
No. of lecture Hours/week	**2**

Sr No.	Course Learning Objectives:
CLO1	Define random experiments, sample space, events, and types of probability
CLO2	Describe properties and applications of various distributions.
CLO3	Apply Theorems and Conditional to solve real-world problems and study of discrete and continuous random variables.
CLO4	Evaluate hypothesis tests, interpret p-values, and validate conclusions.
CLO5	Apply parametric and non-parametric tests and build appropriate statistical models.

Course Outcome

	On completing the course, the student will be able to:
CO1	Recall and define random experiments, sample spaces, events, and probability distributions.
CO2	Describe conditional probability, Bayes' theorem, and distinguish between discrete and continuous variables.
CO3	Solve problems using elementary theorems, Bayes' theorem, and analyze random variables.
CO4	Interpret cumulative distribution functions and assess expectations and variances.
CO5	Analyze hypothesis test results and interpret p-values and confidence intervals.
CO6	Use non-parametric tests for data analysis.

Detailed Syllabus

Module	Title with content	No. of lectures
I	**Probability:** Random experiment, sample space, events types and operations of events, Probability definition: classical, axiomatic, Elementary Theorems of probability (without proof). Conditional probability,,,Bayes‟‟theorem, independence, Examples on Probability. **Random Variables:** Concept and definition of a discrete random variable and continuous random variable. Probability mass function, Probability density function and cumulative distribution function of discrete and continuous random variable, Properties of cumulative distribution function. **Mathematical Expectation and Variance:** Expectation of a function, Variance and S.D of a random variable, properties.	**15**
II	**Standard Probability distributions:** Introduction, properties, examples and applications of each of the following distributions: Binomial distribution, Normal distribution, Chi-square distribution, t distribution, F distribution **Hypothesis testing:** One sided, Two sided hypothesis, critical region, p value, tests based on t, Normal and F, confidence intervals, One-way analysis of variance, two-way analysis of variance. **Non-parametric tests:** Need of non-parametric tests, Sign test, Wilicoxon's signed rank test, run test, Kruskal-Walis tests, Chi square test.	**15**

Textbooks:

1. Gupta, S.C. and Kapoor, V.K. (1987): Fundamentals of Mathematical Statistics, Chand and Sons, New Delhi
2. Goon, A. M., Gupta, M. K. and Dasgupta, B. (1983). Fundamentals of Statistics, Vol. 1, Sixth Revised Edition, The World Press Pvt. Ltd., Calcutta.

References:

1. Mood, A. M. and Graybill, F. A. and Boes D.C. (1974). Introduction to the Theory of Statistics,Ed. 3, McGraw Hill Book Company.
2. Hoel P. G. (1971). Introduction to Mathematical Statistics, John Wiley and Sons, New York.
3. Hogg, R.V. and Craig R.G. (1989). Introduction to Mathematical Statistics, Ed. MacMillan Publishing Co., New York.
4. Walpole R. E., Myers R. H. and Myers S. L. (1985), Probability and Statistics for Engineers and Scientists

Course/ Paper Title	**IOT Technologies**
Course offered as	**Minor II Practical**
Course Code	**RCSIOTMNP304**
Semester	**III**
No. of Credits	**2**
No. of lecture Hours/Week	**2**
No. of Practicals Hours/Week	**2**

Sr No.	**Course Learning Objectives:**
CLO1	Develop the skills to interface Raspberry Pi with external hardware using GPIO and SPI, and program devices like LEDs and cameras
CLO2	Design and implement Python programs to control LEDs, capture images/videos, and program an 8x8 LED grid for various IoT applications
CLO3	Demonstrate the ability to control stepper motors, sensors, and LEDs using Raspberry Pi and implement PWM for device control.
CLO4	Design and deploy IoT systems with Raspberry Pi, using Node-RED, web servers, and sensor integration for real-time data control and interaction

Course Outcome:

	On completing the course, the student will be able to:
CO1	Demonstrate Basic Hardware Setup and GPIO Programming with Raspberry Pi
CO2	Analyze and Implement Communication Protocols for Device Control
CO3	Design and Build IoT Applications Using Node-RED and Web Servers
CO4	Integrate Sensors and Actuators with Raspberry Pi for Advanced Control

Detailed Syllabus

Module	Title with content	No. of lectures
I	**Overview of IoT** Understanding IoT fundamentals, History of IoT, IOT Architecture, protocols **Getting started with Raspberry Pi** Introduction to Raspberry Pi, Comparison of various Rpi Models, Understanding SoC architecture and SoCs used in Raspberry Pi, Pin Description of Raspberry Pi, On-board components of Rpi, Projects using Raspberry Pi, Arduino, atmega328 architecture **Booting Up RPi- Operating System and Linux Commands** Linux- Introduction, Raspbian O.S.- Introduction, Installing Raspbian on Pi, First boot and Basic Configuration of Pi	15
II	**Sensors Interfacing** Temperature and Humidity Sensor (DHT11), Motion Sensor (PIR), Obstacle detection using Ultrasonic sensor, etc, Communicating using RPi-GSM interfacing, Accessing on-board Wi-Fi, Connecting **IoT Design using Raspberry Pi** IoT Applications based on Pi, Node-RED, MQTT Protocol **Interfacing with IoT Platforms:** Basic hardware components like LED, Button, Camera, 8X8 LED Grid, Motor etc and interfacing them for input/output with IoT devices using PWM, GPIO, SPI **Using Sensor & Actuators**: Overview of Sensors working, Interfacing of Actuators, Servo Motor	15

Text Books:

1. Simon Monk , Raspberry Pi Cookbook, O'Reilly, 2014
2. Jain, Prof. Satish, Singh, Shashi, "Internet of Things and its Applications", 1st Edition, BPB, 2020.

Additional References:

1. Internet of Things by Vinayak Shinde, SYBGEN Learning India Pvt. Ltd, 2020
2. Mastering the Raspberry Pi, Warren Gay, Apress, 2014

Sr No.	Minor - IOT Technologies Practicals
1	a) Introduction of Raspberry Pi Kit Styructure b) Preparing Raspberry Pi: Hardware preparation and Installation.
2	GPIO: Light the LED with Python with/without a button using Raspberry Pi. a) Write a program to light a single LED b) Write a program to light a single LED 5 times
3	GPIO: Light the LED with Python with/without a button using Raspberry Pi. a) Write a program in python to light 2 LED simultaneously b) Write a program in python to light 4 LED simultaneously
4	SPI: Camera Connection and capturing Images/Videos using SPI a) Write a program to capture a image with/without text b) Write a program to record video
5	GPIO: LED Grid Module: Program the 8X8 Grid with Different Formulas
6	a) Stepper Motor Control: PWM to manage stepper motor speed using Raspberry Pi. b) Linux Commands.
7	a) Node RED: Connect LED to Internet of Things b) Installing LAMP stack on Raspberry Pi and testing Accessing web application from outside.
8	Practical on demonstration of different types of sensors (LDR, Temperature) with Raspberry Pi.
9	Setting up a web server using Raspberry Pi.
10	Dim a LED using pulse-width modulation with Raspberry Pi.

Course/ Paper Title	**JAVA Programming**
Course offered as	**SEC**
Course Code	**RCSJPSECP306**
Semester	**III**
No. of Credits	**2**
No. of lecture Hours/week	**4**

Sr No.	Course Learning Objectives:
CLO1	Implement OOP concepts effectively in the development of Java-based applications.
CLO2	Develop simple java applications using loops and control statements.
CLO3	Implement the concepts of Method Overloading, Constructor overloading, Static members
CLO4	Analyse and handle errors in java programming using exception handling mechanism.
CLO5	Develop GUI-based desktop applications using Java Swing.
CLO6	Design and deploy dynamic web applications using Servlets and JSP, integrating client-server interactions.

Course Outcome

	On completing the course, the student will be able to:
CO1	Understand and apply basic Java concepts like input/output, loops, conditionals, and arithmetic through programs like multiplication tables and patterns.
CO2	Apply object-oriented programming principles such as inheritance, polymorphism, and encapsulation to design and implement Java programs that solve real-world problems.
CO3	Analyse arrays and errors in java programming and applying appropriate exception handling techniques
CO4	Create Java applications with user interfaces using Swing components and implement Java concepts like JDBC to interact with databases
CO5	Creating servlets and JSPs to handle dynamic data and requests.

CO6	Evaluate and optimize Java programs for efficient database operations (CRUD operations)

Sr No.	Java Programming Practicals
1	Write a Java program that takes a number as input and prints its multiplication table upto 10.
2	Write a Java program to display the following pattern. * * * * * *
3	Find the smallest and largest element from the array.
4	Write a program to create a class and implement the concepts of Method Overloading.
5	Write a program to create static methods and static variables.
6	Write a program to create a class and implement the concepts of Constructor Overloading., Static methods.
7	Write a program to implement the concept of Single level Inheritance.
8	Write a program to implement the concept of Inheritance and Method Overriding.
9	Write a program to implement the concept of interfaces.
10	Write a program to demonstrate the methods of: a. List interface b. Set interface c. Map interface.
11	Write a program to define user defined exceptions and raise them as per the requirements.
12	Write a program using various swing components design Java application to design calculator.
13	Write a JDBC program to insert / update / delete records into a given table
14	Construct a GUI using JAVA Swings to accept Employee details of a record of a given table and submit it to the database using JDBC technology on the click of a button.
15	Write a Servlet that displays the factorial of a number. Accepts a number from the user.
16	Write a Servlet that accepts a User Name from a HTML form and stores it into session. Display the session details using another servlet.

17	Write a Servlet that displays all the records of a table.
18	Write a simple JSP application that demonstrate the use of expression, declaration tags and Sciptlet tags.
19	Write a JSP program to include header and footer pages.
20	Write a registration JSP page that accept the data for a given table and stores it in the database.

SEMESTER IV

Course/ Paper Title	**Theory of Computation**
Course offered as	**Major I**
Course Code	**RUCSMJ401**
Semester	**IV**
No. of Credits	**4**
No. of lecture Hours/week	**4**

Course Learning Objective:

Sr No.	Course Objectives:
	By the end of the course, students will be able to:
CLO1	Understand the fundamental concepts of regular languages, finite automata, and regular expressions.
CLO2	Analyze the equivalence between different types of automata and their conversion methods.
CLO3	Apply concepts of Moore and Mealy machines to model real-time systems with outputs.
CLO4	Evaluate context-free grammars, derivation trees, ambiguity, and simplification techniques and demonstrate the ability to prove properties of languages using formal techniques like the Pumping Lemma and closure properties.
CLO5	Construct Turing Machines for solving decision problems and demonstrate their computational power.
CLO6	Demonstrate the ability to prove properties of languages using formal techniques like the Pumping Lemma and closure properties.

Course Outcome:

	By the end of the course, students will be able to:
CO1	Explain the theoretical foundations of formal languages and automata, including regular and context-free languages.
CO2	Design finite automata, pushdown automata, and Turing machines for specific computational problems.
CO3	Demonstrate the equivalence between various automaton models and evaluate their limitations.
CO4	Apply the pumping lemma and other formal techniques to prove language properties.
CO5	Classify languages based on the Chomsky hierarchy and relate them to appropriate computational models.

Detailed Syllabus

Module	Title with content	No. of lectures
I	**Introduction and Regular Languages: Languages:** Alphabets and Strings. Regular Languages: Regular Expressions, Regular Languages, Regular Grammars, RL and LL grammars, Closure properties. Regular Expressions, Finite automata and Regular Expressions, Closure Properties. **Automata Theory**: Defining Automaton, Finite Automaton, Transitions and Its properties, Acceptability by Finite Automaton, Nondeterministic Finite State Machines, DFA and NDFA equivalence.Equivalence between NFA with and without ε- transitions.	**15**
II	**Finite State Machines with output:** Moore and Mealy machines. Moore and Mealy machine conversion. Limitations of FA. Minimizing Automata. **Formal Languages:** Defining Grammar, Derivations, Languages generated by Grammar, Chomsky Classification of Grammar and Languages, Recursive Enumerable Sets, Operations on Languages, Languages and Automata.	**15**
III	**Context Free Languages:** Context-free Languages, Derivation Tree, Ambiguity of Grammar, CFG simplification, Normal Forms, Pumping Lemma for CFG and its Applications. **Linear Bound Automata:** The Linear Bound Automata Model, Linear Bound Automata and Languages.	**15**
IV	**Pushdown Automata**: Definitions, Acceptance by PDA, PDA and CFG. **Turing Machines:** Turing Machine Definition, Representations, Acceptability by Turing Machines, Designing and Description of Turing Machines, Turing Machine Construction, Variants of Turing Machine.	**15**

References:

1. Theory of Computer Science, K. L. P Mishra, Chandrasekharan, PHI, 3rd Edition 2019
2. Introduction to Computer Theory, Daniel Cohen, Wiley, 2nd Edition, 2007
3. Introductory Theory of Computer Science, E.V.Krishnamurthy, Affiliated East-West Press, 2009
4. Theory of Automata, Languages and Computation Rajendra Kumar. Tata McGraw-Hill

Additional References:

1. Theory of Computation, Kavi Mahesh, Wiley India, 2018
2. Elements of The Theory of Computation, Lewis, Papadimitriou, PHI, 2015
3. Introduction to Languages and the Theory of Computation, John E Martin, McGraw-Hill Education, 2010.
4. Introduction to Theory of Computation, Michel Sipser, Thomson
5. Introduction to Automata Theory, Languages and Computation, John E. Hopcroft, Pearson Education, 2014

Course/ Paper Title	**Computer Networks**
Course offered as	**Major II**
Course Code	**RUCSMJ402**
Semester	**IV**
No. of Credits	**4**
No. of lecture Hours/week	**4**

Sr No.	**Course Learning Objectives**
CLO1	The principles of network communication, including network models (OSI & TCP/IP), physical layer concepts, and transmission media.
CLO2	Apply error detection, Understand on and correction techniques, data link control mechanisms, and VLAN configurations to improve network efficiency.
CLO3	Analyze different network layer protocols, unicast routing techniques, and next-generation IP for effective data transmission.
CLO4	Evaluate wired and wireless LAN architectures, comparing their performance, security, and implementation in real-world scenarios.

Course Outcome

	On completing the course, the student will be able to:
CO1	Explain network models, physical layer functionalities, and transmission media used in data communication.
CO2	Implement error detection and correction methods, data link control mechanisms, and VLAN configurations to enhance network efficiency.
CO3	Examine various network layer protocols, unicast routing strategies, and next-generation IP for efficient data transmission.
CO4	Compare and evaluate wired and wireless LAN architectures based on performance, security, and real-world applications.

Detailed Syllabus

Module	Title with content	No. of lectures
I	**Introduction**: Components, Data Representation, Data Flow, Networks, Network Criteria, Physical Structures, Network Topology, Network types:- LAN, MAN, WAN, Protocols & standards. **Network Models**: The OSI model, TCP/IP protocol suite. **Introduction to Physical layer:** Data and signals, periodic analog signals, Digital signals, transmission impairment, data rate limits, performance. **Analog-to-digital conversion**: PCM,DM, transmission modes. **Analog-to-analog conversion:** AM, FM, PM	**15**
	Bandwidth Utilization – Multiplexing, WDM, FDM, TDM. **Connecting Devices and Virtual LANs**: Hardware Devices used for Networking: Network Interface Card (NIC), Modem, Hub, Switch L1 and L2 switches, Comparison between switch, Bridge, Router, Gateway. **Transmission media:** Guided Media, Unguided Media. **Switching:** Introduction, Circuit Switched Network, Packet Switching.	
II	**Introduction to Data Link Layer:** Link layer addressing, Data Link Layer Design Issues. **Error detection and correction:** Block coding, cyclic codes, checksum, Forward error correction, error correcting codes, error detecting codes. **Data Link Control:** DLC services, HDLC, Point-to-point protocol. **Wired LANs:** Ethernet - Ethernet Protocol, standard Ethernet, fast Ethernet, gigabit Ethernet, 10 gigabit Ethernet **Wireless LANs:** Introduction, IEEE 802.11 project, Bluetooth, WiMAX, Cellular telephony, Satellite networks.	**15**
III	**Introduction to Network Layer**: Network layer services, packet switching, network layer performance, IPv4 addressing, classful and classless addressing forwarding of IP packets. **Network Layer Protocols:** Internet Protocol, ICMPv4, Mobile IP. **Unicast Routing:** Introduction, routing algorithms, unicast routing Protocols, RIP, OSPF, BGP. **Next generation IP:** IPv6 addressing, IPv6 protocol, ICMPv6 protocol, transition from IPv4 to IPv6.	**15**
IV	**Introduction to the Transport Layer**: Transport Layer Protocol, User Datagram Protocol, Transmission Control Protocol, SCTP. **Introduction to Application Layer:** WWW & its architectural overview, HTTP, FTP, Electronic Mail, TELNET, Secure Cell, DNS, SNMP, SMTP. **Quality of Service:** Data Flow to improve QoS, Flow control to improve QoS, Integrated service (Intserv), Differentiated Service (Diffserv).	**15**

References:

1. Data Communications and Networking, Behrouz A. Forouzan, Fifth Edition, TMH, 2018.
2. Computer Network, Andrew S. Tanenbaum, David J. Wetherall, Fifth Edition, Pearson, Education, 2018.

Additional References:

1. Computer Network, Bhushan Trivedi, Oxford University Press, 2016
2. Data and Computer Communication, William Stallings, PHI, 2017

Course/ Paper Title	**MJP4: Computer Science Practical 4**
Course offered as	**Major Practicals**
Course Code	**RUCSMJP4**
Semester	**IV**
No. of Credits	**2**
No. of lecture Hours/week	**4**

Sr No.	Course Learning Objectives:
CLO1	Understand how theoretical concepts like regular expressions, automata, and grammars can be applied programmatically.
CLO2	Implement regular expressions and finite automata for string pattern recognition and validation.
CLO3	Analyze the behavior of PDAs and Turing machines for recognizing complex languages.
CLO4	Develop programs to simulate lexical analysis processes such as tokenization and grammar-based derivations.
CLO5	Design machines (FA, PDA, TM) for specific language constraints and validate their correctness with sample inputs.
CLO6	Understand and execute basic networking to analyze network connectivity and diagnose network issues.
CLO7	Apply network configuration techniques by creating and configuring networks with static and dynamic IP addressing in Cisco Packet Tracer.
CLO8	Design and implement routing protocols in multi-router networks to demonstrate inter-network connectivity.
CLO9	Configure and demonstrate wireless networking using access points and ensure connectivity between multiple devices.
CLO10	Analyze network traffic using Wireshark to capture and filter protocol data and verify the functioning of various protocols.

Course Outcome

	On completing the course, the student will be able to:
CO1	Write programs for tokenization and grammar parsing using regular expressions.
CO2	Construct finite automata that accept specific string patterns (e.g., ending with 101, containing three consecutive 1s).
CO3	Simulate automata and Turing machines for specific language recognition tasks (e.g., equal number of 1’s and 0’s, anbncn).
CO4	Demonstrate the use of pushdown automata to accept palindromes and other context-free languages (e.g., WCWR).
CO5	Analyze and debug automata-based programs to ensure correctness of output with respect to formal language rules.
CO6	Execute basic networking commands and interpret their outputs to troubleshoot and analyze network performance.

CO7	Set up and configure static and dynamic IP addresses in network environments, ensuring successful connectivity.

CO8	Implement and test different routing protocols to establish inter-router communication and connectivity.
CO9	Build and test wireless networks, ensuring proper configuration of access points and device connectivity.
CO10	Capture and analyze network traffic using Wireshark to observe and verify protocol behavior and transactions.

Sr No.	MODULE 1: Theory of Computation Practicals
1	Write a program for tokenization of given input.
2	Write a program for generating regular expressions for regular grammar.
3	Write a program for generating derivation sequence / language for the given sequence of productions.
4	Design a Program for creating machine that accepts three consecutive one.
5	Design a Program for creating machine that accepts the string always ending with 101.
6	Design a program for accepting decimal number divisible by 2.
7	Design a program for creating a machine which accepts string having equal no. of 1's and 0's.
8	Design a program for creating a machine which count number of 1's and 0's in a given string.
9	Design a PDA to accept WCWR where w is any string and WR is reverse of that string and C is a Special symbol.
10	Design a Turing machine that's accepts the following language an b n c n where n>0.

Sr.No	MODULE 2: Computer Networks Practicals
1	Using, linux-terminal or Windows-cmd, execute following networking commands and note the output: ping, traceroute, netstat, arp, ipconfig, Getmac, hostname, NSLookUp, pathping, SystemInfo.
2	Using Packet Tracer, create a basic network of two computers using appropriate network wire. Use Static IP address allocation and show connectivity
3	Using Packet Tracer, create a basic network of One server and two computers using appropriate network wire. Use Dynamic IP address allocation and show connectivity
4	Using Packet Tracer, create a basic network of One server and two computers and two mobile / movable devices using appropriate network wire. Show connectivity
5	Using Packet Tracer, create a network with three routers with RIPv1 and each router associated network will have minimum three PC. Show Connectivity
6	Using Packet Tracer, create a network with three routers with RIPv2 and each router associated network will have minimum three PC. Show Connectivity
7	Using Packet Tracer, create a network with three routers with OSPF and each router associated network will have minimum three PC. Show Connectivity
8	Using Packet Tracer, create a network with three routers with BGP and each router associated network will have minimum three PC. Show Connectivity
9	Using Packet Tracer, create a wireless network of multiple PCs using appropriate access point.

10	Using Wireshark, network analyzer, set the filter for ICMP, TCP, HTTP, UDP, FTP and perform respective protocol transactions to show/prove that the network analyzer is working

Course/ Paper Title	**Fundamental of Data Science**
Course offered as	**Minor I**
Course Code	**RUCSMN403**
Semester	**IV**
No. of Credits	**2**
No. of lecture Hours/week	**2**

Sr No.	Course Learning Objectives:
CLO1	Understand the fundamental concepts, scope, and applications of Data Science across various domains.
CLO2	Identify and classify different types of data and data sources used in Data Science projects.
CLO3	Apply data preprocessing techniques such as data cleaning, transformation, feature selection, and merging.
CLO4	Analyze datasets using descriptive statistics and perform hypothesis testing for data-driven decision making.
CLO5	Create effective data visualizations and communicate insights through data storytelling and visualization tools.

Course Outcome

	On completing the course, the student will be able to:
CO 1	Explain the role and significance of Data Science in real-world applications and how it compares with AI, ML, and BI.
CO 2	Differentiate between structured, semi-structured, and unstructured data, and evaluate suitable data sources for analysis.
CO 3	Perform data preprocessing tasks including handling missing values, encoding, and feature selection using tools like Pandas.
CO 4	Interpret results from statistical methods like mean, median, standard deviation, t-tests, chi-square tests, and ANOVA.
CO 5	Design and implement effective visualizations using tools such as Matplotlib, Seaborn, and Tableau to support data storytelling.

Detailed Syllabus

Module	Title with content	No. of lectures
I	**Introduction:** Definition and scope of Data Science, Applications and domains of Data Science, Comparison with other fields like (BI), Artificial Intelligence (AI), Machine Learning (ML), **Different types of data:** Structured, unstructured, semi-structured, Qualitative & Quantitative, Nominal & Ordinal, Discrete and Continuous Data. **Data sources:** Databases, files, APIs, web scraping, sensors, social media **Data Preprocessing:** Data cleaning: handling missing values, outliers, duplicates, Data transformation: scaling, normalization, encoding categorical variables. **Feature selection**: selecting relevant features/columns. **Data merging**: combining multiple datasets.	**15**
II	**Descriptive statistics**: mean, median, mode, standard deviation, Quartiles. **Hypothesis testing:** t-tests, chi-square test, ANOVA. **Data Visualization techniques:** Principles of effective data visualization, Types of visualizations: bar charts, line charts, scatter plots, histograms, box plots, heat map, ggplot etc. **Visualization tools:** Matplotlib, Seaborn, Tableau, etc. **Data storytelling:** communicating insights through visualizations **Data Management:** Introduction to data management activities, Data pipelines: data extraction, transformation, and loading (ETL), Data governance and data quality assurance, Data privacy and security considerations.	**15**

References:

1. Data Science from Scratch First Principles with Python- Joel Grus O'reilly, 2nd Edition
2. Advancing into Analytics From Excel to Python and R, George Mount, Oreilly, First Edition
3. Introduction to Machine Learning with Python, Andreas C. Muller, Sarah Guido, Oreilly, First Edition

Additional References:

1. Doing Data Science, Rachel Schutt and Cathy O'Neil, O'Reilly,2013
2. Mastering Machine Learning with R, Cory Lesmeister, PACKT Publication,2015
3. Hands-On Programming with R, Garrett Grolemund,1st Edition, 2014
4. An Introduction to Statistical Learning, James, G., Witten, D., Hastie, T., Tibshirani, R.,Springer,2015

Course/ Paper Title	**Advanced Application Development**
Course offered as	**Minor II Practical**
Course Code	**RUCSMNP404**
Semester	**IV**
No. of Credits	**2**
No. of lecture Hours/week	**2**
No. of Practical Hours/week	**2**

Sr No.	Course Learning Objectives:
	On completing the course, the student will be able to:
CLO1	Understand the fundamental concepts and installation procedures of modern web and application development technologies such as MongoDB, Node.js, AngularJS, and Dart.
CLO2	Apply core programming constructs to implement CRUD operations and data handling in MongoDB, Node.js, and Dart environments.
CLO3	Develop dynamic front-end applications using AngularJS by creating modules, controllers, and form validations.
CLO4	Integrate various technologies such as Express.js, MongoDB, and AngularJS to build full-stack web applications.
CLO5	Design and build real-time applications using event-driven and modular programming principles in modern frameworks.

Course Outcome

	On completing the course, the student will be able to:
CO1	Explain the architecture and features of MongoDB, Node.js, AngularJS, and Dart, and install these technologies on a local machine.
CO2	Implement CRUD operations using MongoDB and create collections with appropriate schema and conditions.
CO3	Develop responsive and interactive front-end applications using AngularJS features like directives, controllers, ng-model, and expressions.
CO4	Analyze the interaction between front-end and back-end components in a full-stack application using Node.js, Express, and MongoDB.
CO5	Create modular, event-driven, and scalable web applications, including real-time systems, by integrating multiple frameworks and technologies.

Detailed Syllabus

Module	Title with content	No. of lectures
I	**MongoDB:** Introduction to MongoDB, Connecting to MongoDB. Creating database , collections and documents,Perform CRUD operations in MongoDB, drop collection, drop database. **Node.js** : Introduction to Node.js, installing Node.js, Creating a Sample Node.js Application, Node.js Modules, Node.js HTTP module, Module, Creating an HTTP Web Server with Node.js, Responding to HTTP Requests. **AngularJS:** Introducing AngularJS, Benefits, Basic AngularJS Directives, Nglf Directive. NgForEach Directive, Anatomy of an Angular JS components. **Angular Modules:** Controllers, Creating NgModules One-way and two way Data binding, Validating Form Data.	**15**
II	**Express Framework:** Introduction to the Express Framework. Installing and Testing Express, Creating a Node.js Express App, routers, creating templates Express Middleware Functions. **Dart Programming Basics:** main () Function, Dart Variables, Dart Data Types, Dart Conditional Operators, Control Flow & Loops. **Functions:** Understanding function declaration and usage. Parameters and return types. Function Structure, Creating a Function, Function Returning Expression. **Object-Oriented Programming (OOP) in Dart:** Creating a Class, Adding Methods to Classes, Class - Getters and Setters, Class Inheritance.	**15**

Sr.No	Minor -Advanced Application Development Practicals
1	Introduction to MongoDB and its Installation.
2	Write MongoDB commands to create database Company and collection emp_personal_details with emp_id, emp_name, emp_address, emp_DOB, emp_age, emp_mobile. Insert one/many documents and Display the documents with appropriate conditions.
3	Write a program to implement CRUD operations on MongoDB.
4	Introduction to Node.js and its Installation.
5	Write a program to perform college form validation using AngularJS
6	Write a program to create and implement modules and controllers in Angular JS .
7	Create a simple HTML “Hello World” Project using AngularJS Framework and apply ngcontroller, ng-model and expressions
8	Develop a Shopping List using AngularJS
9	Develop AngularJS Cart application.
10	Develop AngularJS Translate application
11	Develop AngularJS Form application to demonstrate 5 events and validate data.
12	Introduction to Express and Installation.

13	Write a program to create a simple web application using Express, Node JS and Angular JS.
14	Build a real time chat application in Node.js using Express, Mongoose and Socket.io
15	Creating a Simple RESTful Web App with Node.js, Express, and MongoDB.
16	Introduction to DART and Installation.
17	Write a DART program to take input from user. Depending on whether the entered number is even or odd, print out an appropriate message to the user.
18	Write a DART program to take string as input from user and print out whether this string is a palindrome or not.
19	Write a DART program Polymorphism by Method Overriding.
20.	Write a DART program to demonstrate single inheritance and multilevel inheritance.

Course/ Paper Title	**Android Application Development**
Course offered as	**SEC**
Course Code	**RUCSSECP405**
Semester	**IV**
No. of Credits	**2**
No. of lecture Hours/week	**4**

Sr No.	Course Learning Objectives:
	On completing the course, the student will be able to:
CLO1	Understand the fundamental concepts of Android development, including its architecture and components.
CLO2	Design and develop user interfaces using XML layouts and Android UI components.
CLO3	Implement interactive and event-driven mobile applications using Java/Kotlin.
CLO4	Integrate data storage solutions such as Shared Preferences, SQLite, and Firebase in Android applications.
CLO5	Test, debug, and deploy Android applications on various devices and platforms.

Course Outcome

	On completing the course, the student will be able to:
CO1	Develop basic Android applications using Android Studio and XML-based UI components.
CO2	Implement interactive mobile applications using Java/Kotlin.
CO3	Design and develop visually appealing and user-friendly Android applications.
CO4	Utilize data storage techniques, including Shared Preferences, SQLite, and Firebase.
CO5	Test, debug, and deploy Android applications on various platforms and devices.

Sr. No	Android Application Development Practicals
1	Write a program using Kotlin to implement control structures and loops.
2	Write a program using Kotlin to implement object-oriented concepts.
3	Write a program using Kotlin to implement mutable and immutable collections
4	Write a program using Kotlin to implement delegation and extension function.
5	Write a program using Kotlin to implement visibility modifiers.
6	Write a program using Kotlin to implement exception handling.
7	Write a program using Kotlin to implement
8	Create an Android application to display a Hello Android message.
9	Create an Android application to design screens using different layouts and UI including Button, Edittext, Textview, Radio Button etc.
10	Write an android application demonstrating response to event/user interaction for a. Checkbox b. Radio button c. Button d. Spinner
11	Create an application to create Image Flipper and Image Gallery. On click on the image display the information about the image.
12	Create an Android application to demonstrate Shared Preferences.
13	Create an Android application to demonstrate the use of Broadcast listeners.
14	Create an Android application to create and use Services.
15	Create an Android application to demonstrate XML based animation on text.
16	Create an Android application to demonstrate XML based animation on button.
17	Create media player Application in android that plays audio. Implement play ,Pause, Loop features.
18	Create an Android application using Java that add two numbers on click of a button.
19	Create an Android application using Java that demonstrates alert box.
20.	Create an Android application using Java that demonstrates activity lifecycle.

Semester III & IV Examination Pattern

Evaluation for Major Theory Courses (4 Credit Courses)

I. Continuous Internal Evaluation (CIA) - 40 Marks (40%)	II. Semester End Theory Examination (SEE)- 60 Marks (60%)
(i) Mid-Term Class Test – 20 Marks. (ii) Assignment/ Case study/ Presentations – 20 Marks.	A Semester End Theory Examination of Two hours duration for 60 Marks as per the paper pattern given below

Format of Major Theory Question Paper Pattern

Question	Based on	Options	Marks
Q.1	**Module I**	**Any Three out of Six**	**15**
Q.2	**Module II**	**Any Three out of Six**	**15**
Q.3	**Module III**	**Any Three out of Six**	**15**
Q.4	**Module IV**	**Any Three out of Six**	**15**
Total			**60**

- All questions shall be compulsory with internal choice within the questions.
- Each Question may be subdivided into sub questions as a, b, c, d, etc. & Each sub question carries 5 Marks.

Evaluation for Minor Theory Courses (2 Credit Courses)

I. Continuous Internal Evaluation (CIA) - 20 Marks (40%)	II. Semester End Theory Examination (SEE)- 30 Marks (60%)
(i) Mid-Term Class Test – 10 Marks. (ii) Assignment/ Case study/ Presentations – 10 Marks.	A Semester End Theory Examination of one hour duration for 30 Marks as per the paper pattern given below.

Format of Semester End Theory Examination of Minor Papers:

Question	Based on	Options	Marks
Q.1	**Module I**	**Any Two out of Four**	**10**
Q.2	**Module II**	**Any Two out of Four**	**10**
Q.3	**Module I & II**	**Any Two out of Four**	**10**
Total			**30**

- All questions shall be compulsory with internal choice within the questions.
- Each Question may be subdivided into sub questions as a, b, c, d, etc. & Each sub question carries 5 Marks.

Major/Minor Practical Evaluation (50 Marks)

Sr. No.	Practical Assessment	Marks
1	Practical Question 01(Module 1)	15
2	Practical Question 02 (Module 2)	15
3	Viva	10
4	Journal	10

VSC & SEC Practical Evaluation (50 Marks)

Sr. No.		
1	Practical Question 01	20
2	Practical Question 02	20
3	Viva	5
4	Journal	5

- **Duration will be 02 hours.**
- **Certified Journal is compulsory for appearing at the time of practical Exam.**
- **Minimum 80% practical from each module are required to be completed.**
- **The total Evaluation out of 50 marks.**

Name	Affiliation	Sign
Ms. Ritika Lala	Royal College of Arts, Science and Commerce Mira Road, Thane.	
Prof. Vipul Saluja	RD & SH National & SWA Science College Bandra (W), Mumbai.	
Dr. Anita Chaware	SNDT Women's University, Santacruz (W), Mumbai.	
Prof. Ashish Trivedi	Thakur College of Science and Commerce, Kandivali (E), Mumbai.	
Mr. Azhar Shaikh	Peepal Design , Pune	
Ms. Kamlesh Pal	Royal College of Arts, Science and Commerce Mira Road, Thane.	
Ms. Pramitha Santhumayor	Royal College of Arts, Science and Commerce Mira Road, Thane.	
Mr. Hasan Phudinawala	Royal College of Arts, Science and Commerce Mira Road, Thane.	
Mr. Vinay Dubey	Royal College of Arts, Science and Commerce Mira Road, Thane.	
Ms. Simran Hussain	Royal College of Arts, Science and Commerce Mira Road, Thane.	
Ms. Nitya Sinha	Royal College of Arts, Science and Commerce Mira Road, Thane.	
Ms. Anushka Padhye	Royal College of Arts, Science and Commerce Mira Road, Thane.	

ROYAL COLLEGE OF ARTS SCIENCE & COMMERCE DIST. THANE

Principal
ROYAL COLLEGE OF ARTS
SCIENCE & COMMERCE
Penkar Pada, Mira Road,
Dist. Thane, Pin- 401107.