



Shreyarth University

Gujarat Bhavan, Nr M. J. Library, Ashram Road, Ahmedabad – 380 006

NOTIFICATION-AC0703

No.: SU/ACD/AC/NOT/2025/1397

Date: 08/01/2026

Subject: Approval of syllabus of various programmes of School of Computer Science and Applications.

Read following with respect to the subject mentioned above:

- Resolution no. 3,4,5,6,7,8,9 of the 8th meeting of the Board of Studies of the School of Computer Science and Applications dated 11/07/2025. (Attached as Appendix-1)
- Resolution no. 3,4,5,6,7,8,9 of the 9th meeting of the Board of Studies of the School of Computer Science and Applications dated 17/11/2025. (Attached as Appendix-2)
- Resolution No. 9, 10 of the 7th meeting of the Academic Council dated 01/12/2025. (Attached as Appendix-3)

In pursuance of the above mentioned read i and ii, the following decision is notified with regard to the approval of the syllabus of various programmes of School of Computer Science and Applications:

- Resolved that the recommendations of the Board of Studies of School of Computer Science and Applications be accepted and the post-facto approval be given to the new/modified Syllabus & Teaching & Examination Scheme of following programmes and semesters to be made effective from the academic year 2025-26 and onwards as aligned with NEP-2020 as per appendix-3 attached herewith:

Programme	Sem.	New / Modified Syllabus	Other modifications
BCA, Int. B.Sc.-M.Sc. (CA&IT)	V	New	<ul style="list-style-type: none"> • Summer Internship undertaken after Semester IV be reflected in the Grade Sheet of Semester IV with effect from Academic Year 2026–27. • “Software Engineering” shall replace “Network Security”, with effect from Academic Year 2025–26. • “Server Administration” be added as a Value Added Course from the Academic Year 2026–27.
Int. B.Sc.-M.Sc. (CA&IT)	VII	New	---
M. Sc. (IT)	I	New	<p>In Semester III, students will opt for one specialization track from the following:</p> <ol style="list-style-type: none"> 1. Artificial Intelligence & Data Science 2. Full Stack Development 3. Cloud & DevOps Engineering

[Signature]





Shreyarth University

Gujarat Bhavan, Nr M. J. Library, Ashram Road, Ahmedabad – 380 006

			4. Cybersecurity & Ethical Hacking
B.Sc. Cyber Security Cloud Computing and Networking	III	New	---
B. Sc. Animation, VFX & Gaming	V	New	---
MCA	III	New	---
BCA, Int. B.Sc.-M.Sc. (CA&IT)	I, III	Modification	<ul style="list-style-type: none"> • 2-credit summer internship conducted after Semester 2 should be incorporated into the Semester 3 from academic year 2026-27. Students will learn IKS-II online through Swayam portal academic year 2025-26. • The subject 'Soft Skills' from Semester I has been swapped with the subject 'Communication Skills' from Semester II from academic year 2025-26.

- Resolved that the recommendations of the Board of Studies of School of Computer Science and Applications be accepted and the approval be given to the new/modified Syllabus & Teaching & Examination Scheme of following programmes and semesters to be made effective from the academic year 2025-26 and onwards as aligned with NEP-2020 as per appendix-4 attached herewith:

Programme	Sem.	New / Modified Syllabus
BCA, Int. B.Sc.-M.Sc. (CA&IT)	VI	New
Int. B.Sc.-M.Sc. (CA&IT)	VIII	New
M. Sc. (IT)	II	New
B.Sc. Cyber Security Cloud Computing and Networking	IV	New
B. Sc. Animation, VFX & Gaming	VI	New
MCA	IV	New
BCA, Int. B.Sc.-M.Sc. (CA&IT)	II, IV	Modification

Dr. Suresh Sorathia,
Registrar (I/C),
Shreyarth University, Ahmedabad.





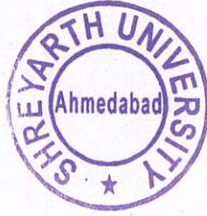
Shreyarth University

Gujarat Bhavan, Nr M. J. Library, Ashram Road, Ahmedabad – 380 006

Attachments: As mentioned above

Copy forwarded for information and necessary action To:

- President, Shreyarth University
- Provost, Shreyarth University
- Library, Shreyarth University
- Director / Head of School of Computer Science and Applications
- All the employees of the School of Computer Science and Applications
- All academic and administrative sections of the University
- Select File





SHREYARTH UNIVERSITY

**School of Computer Science and
Applications**

BCA(H) Semester V

Syllabus

Designed according to NEP 2020

Wef. Year 2025 - 2026

Program Outcomes (POs) – BCA

After completing the BCA program, a graduate will be able to:

1. **Computing Knowledge:** Apply fundamental knowledge of mathematics, computer science, and applications to solve computing problems.
2. **Problem Analysis:** Identify, analyse, and formulate solutions for problems using computer applications and programming concepts.
3. **Design & Development of Solutions:** Design, develop, and evaluate software systems to meet user requirements, considering functional, security, and ethical aspects.
4. **Modern Tool Usage:** Use appropriate programming languages, software tools, and emerging technologies to model and solve real-world problems.
5. **Communication Skills:** Communicate effectively in oral and written forms to share ideas, technical reports, and project documentation with stakeholders.
6. **Teamwork & Leadership:** Function effectively as an individual, and as a member or leader in multidisciplinary teams.
7. **Professional Ethics:** Recognize and commit to professional ethics, responsibilities, and norms of software practices.
8. **Societal Impact:** Understand the impact of computing solutions in a global, economic, environmental, and societal context.
9. **Life-long Learning:** Engage in independent and life-long learning to keep pace with technological advancements in the field of computer applications.
10. **Project Management & Finance:** Apply knowledge of project management principles and computing skills to manage projects efficiently.
11. **Entrepreneurship & Innovation:** Develop entrepreneurial skills and innovative thinking to create solutions, products, or services for real-world needs.

Course: Advance Java								
Course Code:		Credit:4	Semester: V			Programme: BCA(Hons.)		
Teaching Scheme				Evaluation Scheme				
Lecture	Tutorial	Practical	Contact Hours	CIE Theory	CIE Practical	ESE Theory	ESE Practical	Total
3	0	1	5	18/50	-	18/50	-	36/100

Course Outcomes (COs):

- **CO1:** Demonstrate understanding of web architecture, request–response cycle, and REST API fundamentals.
- **CO2:** Apply Java Collections and JDBC programming for effective data handling and persistence.
- **CO3:** Develop and deploy web applications using Servlets and JSP with session and state management.
- **CO4:** Integrate JSON and RESTful APIs for modern web application development.
- **CO5:** Design and implement enterprise-level applications using Spring Boot and Hibernate ORM.
- **CO6:** Collaborate effectively using Git/GitHub and understand modern frontend frameworks conceptually.

Course Learning Outcomes (CLOs)

After completing this course, students will be able to:

1. **Explain** the concepts of web architecture, HTTP/HTTPS protocols, and REST API fundamentals.
2. **Apply** Java Collections and JDBC to perform efficient data handling and database operations.
3. **Develop** dynamic web applications using Servlets and JSP with proper session and state management.
4. **Integrate** JSON and RESTful APIs for data exchange between frontend and backend.
5. **Design and implement** enterprise-level applications using Spring Boot and Hibernate ORM.
6. **Analyze** the role of modern frontend frameworks (React, Angular, Vue) in full-stack development.
7. **Use** Git and GitHub for version control and collaborative project development.
8. **Demonstrate** problem-solving, coding best practices, and deployment skills in Java-based web projects.

Module 1: Web Basics, Collections, and JDBC Essentials

- **Web Architecture:**
 - Client → Server → Database (Request-Response Cycle)
- **HTTP vs HTTPS Basics**
- **REST API Basics:**
 - Conceptual understanding of REST API communication.
- **Java Collections (Basic Recap):**
 - ArrayList, HashMap, Set
- **JDBC Programming:**
 - JDBC Introduction & Drivers
 - Connecting to Database
 - Simple SQL Queries (Select, Insert, Update, Delete)
 - Using Statement & PreparedStatement
 - Basic Transaction Management

Module 2: Servlets and Practical Web Projects

- **Servlet Basics:**
 - Servlet Life Cycle
 - Reading Data from Client (Form Handling)
 - Sending Response to Client
- **Session Management:**
 - HTTP Sessions
 - Cookies Handling
 - URL Rewriting for Session Tracking
- **Practical Projects:**
 - Build a Simple MVC Project (Student Management System)
 - Login System using Servlet + JSP
 - Basic File Upload using Servlet

Module 3: JSP, JSON Integration, and REST API

- **JSP Overview:**
 - JSP Life Cycle
 - JSP Directives, JSP Scripting Elements
 - JSP Implicit Objects
- **JSP Form Processing:**
 - Reading Form Data using JSP
 - Displaying Server Response
- **JSP Session & Cookies Handling**
- **JSP Expression Language (EL):**
 - EL Basics and Implicit Objects
- **JSON Handling:**
 - What is JSON
 - Reading and Sending JSON Data in Java Web Applications
- **REST API Integration:**
 - RESTful API Conceptual Flow
 - How Web Frontend Connects with Backend

Module 4: Spring Boot, Hibernate, and Modern Frontend

- **Spring Boot:**
 - Spring Boot Introduction
 - Annotation-Based Configuration
 - Simple CRUD using Spring Boot + Hibernate
- **Hibernate (Simplified Overview):**
 - What is ORM?
 - Hibernate Annotations
 - Basic CRUD with Hibernate (One Example)
- **Modern Frontend Awareness:**
 - Introduction to React, Angular, Vue (Conceptual Only)
- **Version Control:**
 - Basic Git & GitHub Usage for Web Projects
 -

List of Suggested titles of Experiments:

1. Write a Java program that makes a connection with database using JDBC and insert, Update, delete and display student information from Database.
2. Write a Java program that makes a connection with database using JDBC and prints metadata of this connection and ResultSetMetadata.
3. Create a simple calculator application that demonstrates the use of RMI.
4. Write a Web application using Servlet to find the sum of all the digits of an input integer.
5. Write a Servlet Which Accept Three Number Using Post Method and Display Maximum.
6. Create login form and perform state management using Cookies, HttpSession and URL Rewriting. (Servlet)
7. Write a Web Application Using Filter to print the client's IP address and the current date time, each time it would access any Servlet.
8. Write a Simple application for forwarding data from database to Servlet, Servlet to jsp and Display it Using JSP Scriptlet Tag.
9. Write a Jsp Page That Accept Two Numbers and Find All Prime Number Between Them.
10. Write a Web Application Using Jsp To Demonstrate Login Module.
11. Create database of student subject-wise data and retrieve all data using JSP and generate xml structure along with DTD and XML Schema definition.
12. Refer Practical 11 and apply XSLT (Style) to generated xml document and print your result.
13. Study and implement Hibernate.
14. Study and Implement MVC using Spring Framework.

Text Books:

1. Black Book “Java Server Programming” – J2EE (1st Edition), DreamTech Press, 2008.
2. Complete Reference J2EE by James Keogh, McGraw Hill Education.

Reference Books:

1. Professional Java Server Programming by Subrahmanyam Allamaraju, Cedric Buest, Wiley Publication.
2. SCWCD by Matthew Scarpino, Hanumant Deshmukh, Jignesh Malav, Manning Publication.
3. Core Java, Volume II: Advanced Features by Cay Horstmann and Gary Cornell, Pearson Education.

PO-COMPETENCY-CO MAPPING

CO \ PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	3	2	2	2	1		1	1	2		
CO2	3	3	2	3					2	1	
CO3	3	3	3	3	2	1	1	1	2	1	
CO4	3	2	3	3	1				2		2
CO5	3	3	3	3	1	2	1		2	2	2
CO6	2	2	2	3	3	3	1		3	2	2

Course: Data Communication and Networking									
Course Code:		Credit:4		Semester: V			Programme: BCA(Hons.)		
Teaching Scheme					Evaluation Scheme				
Lecture	Tutorial	Practical	Contact Hours	CIE Theory	CIE Practical	ESE Theory	ESE Practical	Total	
4	0	0	4	18/50	-	18/50	-	36/100	

Course Outcomes (COs):

After completing this course, students will be able to:

- **CO1:** Explain fundamental concepts of data communication, types of networks, topologies, and transmission modes.
- **CO2:** Differentiate between guided and unguided transmission media, and analyze the role of networking devices.
- **CO3:** Compare and contrast OSI and TCP/IP layered architectures, and identify connection-oriented vs. connectionless communication.
- **CO4:** Apply IP addressing schemes, subnetting, and masking in network design.
- **CO5:** Demonstrate understanding of network applications and protocols such as DNS, Email, FTP, and HTTP.

Course Learning Outcomes (CLOs)

At the end of the course, learners will be able to:

1. **Describe** the characteristics and applications of data communication and computer networks.
2. **Classify** and compare different types of transmission media and networking devices.
3. **Illustrate** the layered approach of communication using OSI and TCP/IP models.
4. **Apply** IP addressing, subnetting, and masking to networking problems.
5. **Analyze** the functioning of application layer protocols (DNS, Email, FTP, HTTP) in real-time communication.

Module: I Data Communication: Introduction (25%)

Data Communication: Definition, Characteristics, **Computer Network:** Pros and Cons, Applications, **Types of Area Networks:** LAN, MAN, WAN, **Line Configuration and Its classification,** **Types of Network Topologies:** Bus Topology, Star Topology, Ring Topology, Tree Topology, **Data Flow modes:** Simplex, Half-Duplex, Full-Duplex.

Module: II Transmission Media and Network Devices (25%)

Transmission Media: Definition, Classification, **Guided Media:** Twisted Pair cable, Coaxial cable, Fiber-optic cable, **Unguided Media:** Radio waves, Microwaves, Infrared waves. **Network Devices:** Definition, **Types:** Repeater, Hub, Bridge, Switch: 2-Layer Switch, 3-Layer Switch, Router, Gateways.

Module: III Layered Models (25%)

Network Model Based on Layered Architecture, OSI Model, TCP/IP Model, Connection-oriented and Connectionless Approach, Comparison of OSI Model and TCP/IP Model.

Module: IV IP Protocol and Network Applications (25%)

IP Protocol, Addressing Schemes, Subnet and Masking, DNS, Email Protocols, FTP, HTTP.

Text Books:

1. Fourouzan Behrouz A., Data Communication and Networking, 3rd Edition, Tata McGraw Hill Education Private Limited.
2. ISRD Group, Data Communication and Computer Networks.

Reference Books:

1. Tomasi Wayne, Introduction to Data communications and Networking, Pearson.
2. Zheng Youlu and Akhtar Shakil, Networks for Computer Scientists and Engineering, Oxford.
3. Godbole Achyut and Kahate Atul, Data Communications and Networks, 2nd Edition, Tata McGraw Hill Education Private Limited.

PO-COMPETENCY-CO MAPPING

CO \ PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	3	2	2	2	1		1	1	2		
CO2	3	3	2	3			1		2	1	
CO3	3	3	3	3	1		1	1	2		
CO4	3	3	3	3					2	2	2
CO5	3	2	3	2	2	1	1	1	2		2

Course: Web Application Development using PHP and MYSQL								
Course Code:		Credit:4		Semester: V			Programme: BCA	
Teaching Scheme				Evaluation Scheme				
Lecture	Tutorial	Practical	Contact Hours	CIE Theory	CIE Practical	ESE Theory	ESE Practical	Total
3	0	1	5	18/50	-	18/50	-	36/100

Course Outcomes (COs):

After completing this course, the student will be able to:

- **CO1:** Explain PHP environment setup, syntax, and integration with HTML.
- **CO2:** Apply control structures, arrays, and user-defined functions in PHP programs.
- **CO3:** Develop dynamic web pages using PHP form handling, validation, sessions, cookies, and file uploads.
- **CO4:** Implement database-driven applications by integrating PHP with MySQL.
- **CO5:** Demonstrate practical skills in building end-to-end web applications using PHP & MySQL.

Course Learning Outcomes (CLOs)

By the end of this course, students will be able to:

1. **Set up and configure** PHP, MySQL, and web server environments for application development.
2. **Apply** PHP syntax, variables, arrays, functions, and control structures to develop modular code.
3. **Design and validate** interactive web forms with secure handling of user input.
4. **Implement** state management techniques using sessions and cookies in dynamic applications.
5. **Integrate** PHP with MySQL to perform database connectivity and CRUD operations.
6. **Develop and test** database-driven web applications with features such as file uploads, email, and basic image processing.
7. **Demonstrate** the ability to build complete web-based solutions using PHP and MySQL.

Course Content

Module I: Working with Basic Building Blocks of PHP (25%)

This module introduces PHP, including the installation of Apache, MySQL, and PHP. It covers PHP syntax, integration with HTML, and use of constants and variables. Students will learn to pass variables using URLs, sessions, cookies, and forms. It emphasizes the use of includes, alternate syntaxes, and fundamental coding practices.

Module II: Working with PHP Arrays and Functions (25%)

The module covers control structures such as conditional statements and loops, function creation with parameters and return values, and extensive use of arrays, including associative and multidimensional arrays. It also introduces array-related functions.

Module III: Handling HTML Forms with PHP (25%)

This module delves into strings, date and time functions, and form handling using GET and POST methods. It also includes form validations, file uploads, sessions, cookies, and image processing. Students will learn how to send emails using PHP.

Module IV: PHP and MySQL Programming (25%)

This section focuses on connecting PHP with MySQL. It includes creating and managing databases, performing CRUD operations, and using PHP functions like `mysql_connect`, `mysql_query`, and `mysql_fetch_array` to access and manipulate data.

Text Books:

1. Boronczyk Timothy, Naramore Elizabeth, Gerner Jason, Beginning PHP6, Apache, MySQL, Wrox.

Reference Books:

1. Holzner Steven, The Complete Reference PHP, McGraw Hill
2. Suehring Steve, Converse Tim, and Park Joyce, PHP6 and MySQL Bible, WILEY Publication

List of suggestive programs.

1. Create a web page that will print Hello World using PHP script.
2. Create a web page that will demonstrate the use of comments in PHP.
3. Create a web page that will have two variable, one variable will store numeric value and another will store string. Print both values.
4. Create a web page that will generate random number between 1 to 100 using `rand()` function.
5. Create a web page that will generate any random number and find square root of that number using `sqrt()` function.
6. Create a Web-based version of your resume, incorporating headings, lists, and varying text styles.
7. Build a page that calls the `phpInfo()` function and run it from your Web server.
8. Write a Web page that asks the user for his first and last name and then uses a PHP script to write a form letter to that person.
9. Write your own story game. Find or write some text to modify, create an appropriate input form, and output the story with a PHP script.
10. Create a form for the page caption, background color, font color, and text body. Use this form to generate an HTML page.
11. Create a web application that will have array of fruits. Display the fruits name using for each loop.
12. Create a web application that will have array of animals. Make comparison of animals on the basis of their height and display who is highest using operators.

13. Create a PHP script that uses function to test whether a number is greater than 30, 20 or 10 using ternary operator.
14. Create a PHP script using a for loop to add all the integers between 0 and 30 and display the total.
15. Create a PHP script to calculate and print the factorial of a number using a for loop.
16. Create a PHP script using a for loop to add all the integers between 0 and 30 and display the sum.
17. Create a PHP script which display the capital and country name from the array like "Italy"=>"Rome", "United Kingdom"=>"London". Sort the list by the name of the country.
18. Create a PHP script using a function to calculate the factorial of a number (non-negative integer). The function accepts the number as a argument.
19. Write a PHP script to get the first element of the below array. \$color = array(4 => 'white', 6 => 'green', 11=> 'red'); Output: white
20. Write a PHP script to calculate and display average temperature, five lowest temperatures.
21. Create a PHP script using a function to check a number is prime or not.
22. Create a PHP program which iterates the integers from 1 to 100. For multiples of three print "Fizz" instead of the number and for the multiples of five print "Buzz". For numbers which are multiples of both three and five print "FizzBuzz".
23. Create a web application form which includes nickname, first name, memo as field. Ask the user to enter the values and display the same on the web page.
24. Create a simple web application in HTML form and accept the user's name and display the name through PHP echo statement.
25. Write a simple PHP program to check that emails are valid.
26. Write a PHP script to count number of lines in a file.
27. Create a PHP script that demonstrate whether cookie is enabled or not.
28. Create a PHP script that has two session variable called color and animal. Set the value of both session variable and access the session information on another page and print it.
29. Create a PHP script that has one session variable called counter which increments its value as many times user visits the page.
30. Create a PHP script that will ask user to upload a image file in a folder and checks whether it is successfully uploaded or not.
31. Create a web page for simple Login and Logout using sessions.
32. Create a web page that will ask the user to input email address and give validation for proper email address.
33. Write a PHP script to connect the server and database.
34. Create a PHP web page that will ask the user to insert records to the table in Database.
35. Create a PHP web page that will fetch records from the table in Database.
36. Create a PHP web page that will upload image and read the image from the database and display on the web page.
37. Create a web form for Registration with required fields and ask user to input the data and display the same in the table.

38. Create a web form for storing car details. Take all the necessary fields like model no, price, company etc and ask the user to input the details of car and submit it to store into database.
39. Create a web page of feedback with required fields like username, comments etc and ask user to give the feedback and store it in the table in the database.
40. Create a web page that will display all the details of order received and allows user to update the order and store it in a table.
41. Create a Contact us web page with required fields like name, email_id, message etc and ask user to enter the details and store it in a database.
42. Create a web page that will ask user to select any one option whether user wants to eat punjabi, chinese, italian or mexican. Give choices using radio button, when user selects any cuisine display in the message box and store it into the table.

PO-COMPETENCY-CO MAPPING

CO \ PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	3	2	2	3	1				2		
CO2	3	3	2	3	1				2	1	
CO3	3	3	3	3	2	1	1		2	1	
CO4	2	2	3	3	1	1			2		
CO5	3	3	3	3	2	2	1		3	2	2

Course: Linux Shell Scripting								
Course Code:		Credit:4	Semester: V			Programme: BCA		
Teaching Scheme				Evaluation Scheme				
Lecture	Tutorial	Practical	Contact Hours	CIE Theory	CIE Practical	ESE Theory	ESE Practical	Total
3	0	1	5	18/50	-	18/50	-	36/100

Course Outcomes (COs):

By the end of this course, students will be able to:

- CO1.** Explain the architecture, features, file system, and basic commands of Unix/Linux.
- CO2.** Apply general-purpose utilities and file management commands for handling Unix files and directories.
- CO3.** Develop shell scripts using variables, decision-making, and looping constructs.
- CO4.** Demonstrate file handling operations including permissions, ownership, and filters.
- CO5.** Use regular expressions, grep, egrep, and sed for pattern matching and text processing.

Course Learning Outcomes (CLOs):

Having completed this course, the learner will be able to:

1. **Understand** the Unix architecture, features, and command structure.
2. **Execute** general-purpose commands for file, directory, and system management.
3. **Develop and debug** shell scripts for automation using control structures and looping.
4. **Apply** file permission, ownership, and redirection concepts to manage resources securely.
5. **Utilize** filters, pipes, and regular expressions to process and manipulate data efficiently.
6. **Integrate** Unix/Linux commands and shell scripting to solve real-world system-level problems.

Course Content

Module: I Introduction (30%)

Unix Architecture, Features of Unix, Command structure, General Purpose Utilities: Cal, date, echo, bc, who, uname, tty, man, passwd, wc, Linux Files and Directories: Files and its types, Unix file system, absolute and relative file pathnames, Current Working Directory- pwd, Navigating the file system-cd command, HOME variable and Home directory, Listing Files and Directories-ls with options, Methods to create a file- touch, cat, Showing the contents of a file-cat, more, less, Creating Directories-mkdir, Removing Empty Directories- rmdir, Copying Files and Directories-cp, Removing Files and Nonempty Directories-rm, Renaming Files and Directories-mv.

Module: II Shell Script (30%)

Basics of Shell Scripting Programming: Interactive shell script using read and echo; Decision Statements: if then fi, if then else fi, if then elif else fi, case esac; Test command Logical Operators; Looping statements: for loop, while loop, until loop; Break, continue command; expr, Positional parameters and shift operator.

Module: III File Operations (30%)

File Permissions, changing file permissions-chmod, Absolute and Relative file permissions, Directory permissions, File ownership and changing file ownership, Locating files-find. Escaping, quoting, Piping, Redirection, tee, matching file names with patterns (wildcard characters). Simple Filters: Paginating files -pr, Comparing Two Files-cmp, what is Common-comm, Converting One File to Other-diff, head, tail, Splitting a file vertically-cut, Pasting files-paste, Ordering a file-sort, Locate repeated and non-repeated lines-uniq, Translating characters-tr.

Module: IV Filter System (10%)s

Filters using regular expression: Searching for pattern-grep, grep options, Extended Regular Expressions (ERE) and egrep, stream editor -sed, sed options.

Text Books:

1. Sumitabha Das, Unix: Concepts and Applications, Tata McGrawHill, Latest Edition.
2. M.G.Venkateshmurthy, Introduction to UNIX and Shell Programming, Pearson Education, Latest Edition.

Reference Books:

1. Yashvant P Kanetkar, UNIX Shell Programming, BPB Publications, Latest Edition.
2. Christopher Diaz, Introduction to Unix/Linux, Cengage Learning, Latest Edition.
3. B.M. Harwani, Unix and Shell Programming, Oxford University Press, Latest Edition.

List of suggestive programs.

1. Write a shell script that takes a filename as an argument and checks if the file exists and is executable. If the file is executable then the shell script should display the message: "File exists". If the file does not exist and is not executable then the script should display the message: "File does not exist or is not executable."
2. Write a shell script that takes a filename from the user and checks whether it is a directory file or not. If it is a directory, then the script should display the contents of the directory. If it is not a directory file then script should display message: "File is not a directory file "

3. Write a shell script that accepts 2 filenames and checks if both exists; if both exist then append the content of the second file into the first file
4. Write a shell script to check whether the file is read only or not.
5. Write a shell script to display "Good Morning/Good Afternoon/Good Evening/Good Night" according to the current login time.
6. Write a shell script to find whether a given file is zero sized or not, if it is deleted.
7. Write a shell script to enter 3 file names and find the largest of three files.
8. Write a shell script that takes the name of two files from user and performs the following:
 - i. Displays the message: "Displaying the contents of file :(first file)" and displays the contents page wise.
 - ii. Copies the contents of the first file to the second file.
 - iii. Finally displays the message: "File copied successfully."
9. Write a shell script to display last five lines from a file
10. Write a shell script to display the first five lines from a file.
11. Write a shell script to create a file student with following fields: Roll No. Name Address Marks Cut the first two fields from the student and paste it to a new file.
12. Write a shell script to check whether a particular named user is currently logged in or not.

PO-COMPETENCY-CO MAPPING

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

Course: Software Engineering								
Course Code: BCA51CC03			Semester: V			Programme: BCA		
Teaching Scheme				Evaluation Scheme				
Lecture	Tutorial	Practical	Contact Hours	CIE Theory	CIE Practical	ESE Theory	ESE Practical	Total
2	0	0	2	18/50	-	18/50	-	36/100

Course Outcomes (COs)

After completion of this course, students will be able to:

- **CO1:** Explain the fundamentals of software, its types, characteristics, myths, and role of Software Engineering.
- **CO2:** Compare and apply various software process models, including traditional and agile approaches.
- **CO3:** Apply requirements engineering techniques and design principles to develop structured SRS and basic system design.
- **CO4:** Apply software testing methods and quality assurance principles to ensure reliable and standard-compliant software.

Course Learning Outcomes (CLOs):

After successful completion of this course, a student will be able to:

1. **Explain** the fundamental concepts of software engineering, including the nature of software, its types, characteristics, myths, and professional ethics.
2. **Compare** and **analyze** different software process models (traditional and agile) and **apply** suitable methodologies for software development.
3. **Apply** requirements engineering techniques to **capture**, **specify**, and **document** functional and non-functional requirements in the form of an SRS.
4. **Design** software solutions using architectural models, component-level design, and user-interface design principles.
5. **Apply** software testing strategies, **design** test cases using black-box and white-box techniques, and **evaluate** software quality using standards and reviews.

Course Content

Module I: Introduction to Software Engineering (25%)

Definition and Nature of Software, Characteristics of Good Software, Types of Software: System, Application, Embedded, Web.

Introduction to Software Engineering, Importance of Software Engineering in Modern Computing, Software Engineering Ethics and Professional Practice, Software Myths and Realities

Module: II Software Process Models (25%)

Software Process: Definition and Activities (Communication, Planning, Modeling, Construction, Deployment), Prescriptive Process Models: Linear Sequential (Waterfall), Prototyping, RAD, Evolutionary Models: Incremental, Spiral, Agile Software Development: Principles and Practices, Overview of Scrum and Extreme Programming (XP), Comparison of Traditional and Agile Models

Module III: Requirements and Software Design (25%)

Requirements Engineering: Importance and Activities, Functional vs. Non-Functional Requirements, Requirement Elicitation Techniques (Interviews, Questionnaires), Requirements Specification Document (SRS): Structure and Sample, Basic Design Principles and Concepts, Architectural Design: Layered and Client-Server Architecture, Component-Level Design and Simple UI Design Principles

Module IV: Software Testing and Quality Overview (25%)

Importance of Testing in SDLC, Testing Levels: Unit, Integration, System, Acceptance, Testing Techniques: White-box (statement & branch coverage), Black-box (equivalence partitioning, boundary value analysis), Basic Test Case Design, Introduction to Software Quality Assurance (SQA), Software Review Techniques (Walkthroughs, Inspections), Quality Standards Overview: ISO 9000, CMM Levels

Text Books:

1. Pressman Roger S., Software Engineering – A Practitioner's Approach, 8th Edition TATA McGraw Hill Publications.

Reference Books:

1. Sommerville, Software Engineering, 8th Edition Pearson Education.
2. Jawadekar Waman S., Software Engineering – Principles and Practices, TMGH Publication
3. Mall Rajib, Fundamentals of Software Engineering, Prentice-Hall, 2011.
4. Mishra Jibitesh and Mohanty Ashok, Software Engineering, PERSON
5. Datta Subhajit, Software Engineering Concept and Application, OXFORD

6. Jalote Pankaj, Software Engineering – A Precise Approach, Wiley India
7. Jawadekar Waman S., Software Engineering – A Primer, TMGH Publication
8. Pfleeger Shari Lawrence and Atlee Joanne M., Software Engineering – Theory and Practice, 3rd Edition, Pearson Education.

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



SHREYARTH UNIVERSITY

**School of Computer Science and
Applications**

Integrated BSc – MSc (CA & IT)

Semester V

Syllabus

Designed according to NEP 2020

Wef. Year 2025 - 2026

Program Outcomes (POs) – IMSc(CA&IT)

After completing the IMSc(CA&IT) program, a student will be able to:

1. **Computing Knowledge:** Apply fundamental knowledge of mathematics, computer science, and applications to solve computing problems.
2. **Problem Analysis:** Identify, analyse, and formulate solutions for problems using computer applications and programming concepts.
3. **Design & Development of Solutions:** Design, develop, and evaluate software systems to meet user requirements, considering functional, security, and ethical aspects.
4. **Modern Tool Usage:** Use appropriate programming languages, software tools, and emerging technologies to model and solve real-world problems.
5. **Communication Skills:** Communicate effectively in oral and written forms to share ideas, technical reports, and project documentation with stakeholders.
6. **Teamwork & Leadership:** Function effectively as an individual, and as a member or leader in multidisciplinary teams.
7. **Professional Ethics:** Recognize and commit to professional ethics, responsibilities, and norms of software practices.
8. **Societal Impact:** Understand the impact of computing solutions in a global, economic, environmental, and societal context.
9. **Life-long Learning:** Engage in independent and life-long learning to keep pace with technological advancements in the field of computer applications.
10. **Project Management & Finance:** Apply knowledge of project management principles and computing skills to manage projects efficiently.
11. **Entrepreneurship & Innovation:** Develop entrepreneurial skills and innovative thinking to create solutions, products, or services for real-world needs.

Course: Advance Java								
Course Code:		Credit:4	Semester: V			Programme: BCA(Hons.)		
Teaching Scheme				Evaluation Scheme				
Lecture	Tutorial	Practical	Contact Hours	CIE Theory	CIE Practical	ESE Theory	ESE Practical	Total
3	0	1	5	18/50	-	18/50	-	36/100

Course Outcomes (COs):

- **CO1:** Demonstrate understanding of web architecture, request–response cycle, and REST API fundamentals.
- **CO2:** Apply Java Collections and JDBC programming for effective data handling and persistence.
- **CO3:** Develop and deploy web applications using Servlets and JSP with session and state management.
- **CO4:** Integrate JSON and RESTful APIs for modern web application development.
- **CO5:** Design and implement enterprise-level applications using Spring Boot and Hibernate ORM.
- **CO6:** Collaborate effectively using Git/GitHub and understand modern frontend frameworks conceptually.

Course Learning Outcomes (CLOs)

After completing this course, students will be able to:

1. **Explain** the concepts of web architecture, HTTP/HTTPS protocols, and REST API fundamentals.
2. **Apply** Java Collections and JDBC to perform efficient data handling and database operations.
3. **Develop** dynamic web applications using Servlets and JSP with proper session and state management.
4. **Integrate** JSON and RESTful APIs for data exchange between frontend and backend.
5. **Design and implement** enterprise-level applications using Spring Boot and Hibernate ORM.
6. **Analyze** the role of modern frontend frameworks (React, Angular, Vue) in full-stack development.
7. **Use** Git and GitHub for version control and collaborative project development.
8. **Demonstrate** problem-solving, coding best practices, and deployment skills in Java-based web projects.

Module 1: Web Basics, Collections, and JDBC Essentials

- **Web Architecture:**
 - Client → Server → Database (Request-Response Cycle)
- **HTTP vs HTTPS Basics**
- **REST API Basics:**
 - Conceptual understanding of REST API communication.
- **Java Collections (Basic Recap):**
 - ArrayList, HashMap, Set
- **JDBC Programming:**
 - JDBC Introduction & Drivers
 - Connecting to Database
 - Simple SQL Queries (Select, Insert, Update, Delete)
 - Using Statement & PreparedStatement
 - Basic Transaction Management

Module 2: Servlets and Practical Web Projects

- **Servlet Basics:**
 - Servlet Life Cycle
 - Reading Data from Client (Form Handling)
 - Sending Response to Client
- **Session Management:**
 - HTTP Sessions
 - Cookies Handling
 - URL Rewriting for Session Tracking
- **Practical Projects:**
 - Build a Simple MVC Project (Student Management System)
 - Login System using Servlet + JSP
 - Basic File Upload using Servlet

Module 3: JSP, JSON Integration, and REST API

- **JSP Overview:**
 - JSP Life Cycle
 - JSP Directives, JSP Scripting Elements
 - JSP Implicit Objects
- **JSP Form Processing:**
 - Reading Form Data using JSP
 - Displaying Server Response
- **JSP Session & Cookies Handling**
- **JSP Expression Language (EL):**
 - EL Basics and Implicit Objects
- **JSON Handling:**
 - What is JSON
 - Reading and Sending JSON Data in Java Web Applications
- **REST API Integration:**
 - RESTful API Conceptual Flow
 - How Web Frontend Connects with Backend

Module 4: Spring Boot, Hibernate, and Modern Frontend

- **Spring Boot:**
 - Spring Boot Introduction
 - Annotation-Based Configuration
 - Simple CRUD using Spring Boot + Hibernate
- **Hibernate (Simplified Overview):**
 - What is ORM?
 - Hibernate Annotations
 - Basic CRUD with Hibernate (One Example)
- **Modern Frontend Awareness:**
 - Introduction to React, Angular, Vue (Conceptual Only)
- **Version Control:**
 - Basic Git & GitHub Usage for Web Projects
 -

List of Suggested titles of Experiments:

1. Write a Java program that makes a connection with database using JDBC and insert, Update, delete and display student information from Database.
2. Write a Java program that makes a connection with database using JDBC and prints metadata of this connection and ResultSetMetadata.
3. Create a simple calculator application that demonstrates the use of RMI.
4. Write a Web application using Servlet to find the sum of all the digits of an input integer.
5. Write a Servlet Which Accept Three Number Using Post Method and Display Maximum.
6. Create login form and perform state management using Cookies, HttpSession and URL Rewriting. (Servlet)
7. Write a Web Application Using Filter to print the client's IP address and the current date time, each time it would access any Servlet.
8. Write a Simple application for forwarding data from database to Servlet, Servlet to jsp and Display it Using JSP Scriptlet Tag.
9. Write a Jsp Page That Accept Two Numbers and Find All Prime Number Between Them.
10. Write a Web Application Using Jsp To Demonstrate Login Module.
11. Create database of student subject-wise data and retrieve all data using JSP and generate xml structure along with DTD and XML Schema definition.
12. Refer Practical 11 and apply XSLT (Style) to generated xml document and print your result.
13. Study and implement Hibernate.
14. Study and Implement MVC using Spring Framework.

Text Books:

1. Black Book “Java Server Programming” – J2EE (1st Edition), DreamTech Press, 2008.
2. Complete Reference J2EE by James Keogh, McGraw Hill Education.

Reference Books:

1. Professional Java Server Programming by Subrahmanyam Allamaraju, Cedric Buest, Wiley Publication.
2. SCWCD by Matthew Scarpino, Hanumant Deshmukh, Jignesh Malav, Manning Publication.
3. Core Java, Volume II: Advanced Features by Cay Horstmann and Gary Cornell, Pearson Education.

PO-COMPETENCY-CO MAPPING

CO \ PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	3	2	2	2	1		1	1	2		
CO2	3	3	2	3					2	1	
CO3	3	3	3	3	2	1	1	1	2	1	
CO4	3	2	3	3	1				2		2
CO5	3	3	3	3	1	2	1		2	2	2
CO6	2	2	2	3	3	3	1		3	2	2

Course: Data Communication and Networking								
Course Code:		Credit:4		Semester: V			Programme: BCA(Hons.)	
Teaching Scheme				Evaluation Scheme				
Lecture	Tutorial	Practical	Contact Hours	CIE Theory	CIE Practical	ESE Theory	ESE Practical	Total
4	0	0	4	18/50	-	18/50	-	36/100

Course Outcomes (COs):

After completing this course, students will be able to:

- **CO1:** Explain fundamental concepts of data communication, types of networks, topologies, and transmission modes.
- **CO2:** Differentiate between guided and unguided transmission media, and analyze the role of networking devices.
- **CO3:** Compare and contrast OSI and TCP/IP layered architectures, and identify connection-oriented vs. connectionless communication.
- **CO4:** Apply IP addressing schemes, subnetting, and masking in network design.
- **CO5:** Demonstrate understanding of network applications and protocols such as DNS, Email, FTP, and HTTP.

Course Learning Outcomes (CLOs)

At the end of the course, learners will be able to:

1. **Describe** the characteristics and applications of data communication and computer networks.
2. **Classify** and compare different types of transmission media and networking devices.
3. **Illustrate** the layered approach of communication using OSI and TCP/IP models.
4. **Apply** IP addressing, subnetting, and masking to networking problems.
5. **Analyze** the functioning of application layer protocols (DNS, Email, FTP, HTTP) in real-time communication.

Module: I Data Communication: Introduction (25%)

Data Communication: Definition, Characteristics, **Computer Network:** Pros and Cons, Applications, **Types of Area Networks:** LAN, MAN, WAN, **Line Configuration and Its classification,** **Types of Network Topologies:** Bus Topology, Star Topology, Ring Topology, Tree Topology, **Data Flow modes:** Simplex, Half-Duplex, Full-Duplex.

Module: II Transmission Media and Network Devices (25%)

Transmission Media: Definition, Classification, **Guided Media:** Twisted Pair cable, Coaxial cable, Fiber-optic cable, **Unguided Media:** Radio waves, Microwaves, Infrared waves. **Network Devices:** Definition, **Types:** Repeater, Hub, Bridge, Switch: 2-Layer Switch, 3-Layer Switch, Router, Gateways.

Module: III Layered Models (25%)

Network Model Based on Layered Architecture, OSI Model, TCP/IP Model, Connection-oriented and Connectionless Approach, Comparison of OSI Model and TCP/IP Model.

Module: IV IP Protocol and Network Applications (25%)

IP Protocol, Addressing Schemes, Subnet and Masking, DNS, Email Protocols, FTP, HTTP.

Text Books:

1. Fourouzan Behrouz A., Data Communication and Networking, 3rd Edition, Tata McGraw Hill Education Private Limited.
2. ISRD Group, Data Communication and Computer Networks.

Reference Books:

1. Tomasi Wayne, Introduction to Data communications and Networking, Pearson.
2. Zheng Youlu and Akhtar Shakil, Networks for Computer Scientists and Engineering, Oxford.
3. Godbole Achyut and Kahate Atul, Data Communications and Networks, 2nd Edition, Tata McGraw Hill Education Private Limited.

PO-COMPETENCY-CO MAPPING

CO \ PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	3	2	2	2	1		1	1	2		
CO2	3	3	2	3			1		2	1	
CO3	3	3	3	3	1		1	1	2		
CO4	3	3	3	3					2	2	2
CO5	3	2	3	2	2	1	1	1	2		2

Course: Web Application Development using PHP and MYSQL								
Course Code:		Credit:4		Semester: V			Programme: BCA	
Teaching Scheme				Evaluation Scheme				
Lecture	Tutorial	Practical	Contact Hours	CIE Theory	CIE Practical	ESE Theory	ESE Practical	Total
3	0	1	5	18/50	-	18/50	-	36/100

Course Outcomes (COs):

After completing this course, the student will be able to:

- **CO1:** Explain PHP environment setup, syntax, and integration with HTML.
- **CO2:** Apply control structures, arrays, and user-defined functions in PHP programs.
- **CO3:** Develop dynamic web pages using PHP form handling, validation, sessions, cookies, and file uploads.
- **CO4:** Implement database-driven applications by integrating PHP with MySQL.
- **CO5:** Demonstrate practical skills in building end-to-end web applications using PHP & MySQL.

Course Learning Outcomes (CLOs)

By the end of this course, students will be able to:

1. **Set up and configure** PHP, MySQL, and web server environments for application development.
2. **Apply** PHP syntax, variables, arrays, functions, and control structures to develop modular code.
3. **Design and validate** interactive web forms with secure handling of user input.
4. **Implement** state management techniques using sessions and cookies in dynamic applications.
5. **Integrate** PHP with MySQL to perform database connectivity and CRUD operations.
6. **Develop and test** database-driven web applications with features such as file uploads, email, and basic image processing.
7. **Demonstrate** the ability to build complete web-based solutions using PHP and MySQL.

Course Content

Module I: Working with Basic Building Blocks of PHP (25%)

This module introduces PHP, including the installation of Apache, MySQL, and PHP. It covers PHP syntax, integration with HTML, and use of constants and variables. Students will learn to pass variables using URLs, sessions, cookies, and forms. It emphasizes the use of includes, alternate syntaxes, and fundamental coding practices.

Module II: Working with PHP Arrays and Functions (25%)

The module covers control structures such as conditional statements and loops, function creation with parameters and return values, and extensive use of arrays, including associative and multidimensional arrays. It also introduces array-related functions.

Module III: Handling HTML Forms with PHP (25%)

This module delves into strings, date and time functions, and form handling using GET and POST methods. It also includes form validations, file uploads, sessions, cookies, and image processing. Students will learn how to send emails using PHP.

Module IV: PHP and MySQL Programming (25%)

This section focuses on connecting PHP with MySQL. It includes creating and managing databases, performing CRUD operations, and using PHP functions like `mysql_connect`, `mysql_query`, and `mysql_fetch_array` to access and manipulate data.

Text Books:

1. Boronczyk Timothy, Naramore Elizabeth, Gerner Jason, Beginning PHP6, Apache, MySQL, Wrox.

Reference Books:

1. Holzner Steven, The Complete Reference PHP, McGraw Hill
2. Suehring Steve, Converse Tim, and Park Joyce, PHP6 and MySQL Bible, WILEY Publication

List of suggestive programs.

1. Create a web page that will print Hello World using PHP script.
2. Create a web page that will demonstrate the use of comments in PHP.
3. Create a web page that will have two variable, one variable will store numeric value and another will store string. Print both values.
4. Create a web page that will generate random number between 1 to 100 using `rand()` function.
5. Create a web page that will generate any random number and find square root of that number using `sqrt()` function.
6. Create a Web-based version of your resume, incorporating headings, lists, and varying text styles.
7. Build a page that calls the `phpInfo()` function and run it from your Web server.
8. Write a Web page that asks the user for his first and last name and then uses a PHP script to write a form letter to that person.
9. Write your own story game. Find or write some text to modify, create an appropriate input form, and output the story with a PHP script.
10. Create a form for the page caption, background color, font color, and text body. Use this form to generate an HTML page.
11. Create a web application that will have array of fruits. Display the fruits name using for each loop.
12. Create a web application that will have array of animals. Make comparison of animals on the basis of their height and display who is highest using operators.

13. Create a PHP script that uses function to test whether a number is greater than 30, 20 or 10 using ternary operator.
14. Create a PHP script using a for loop to add all the integers between 0 and 30 and display the total.
15. Create a PHP script to calculate and print the factorial of a number using a for loop.
16. Create a PHP script using a for loop to add all the integers between 0 and 30 and display the sum.
17. Create a PHP script which display the capital and country name from the array like "Italy"=>"Rome", "United Kingdom"=>"London". Sort the list by the name of the country.
18. Create a PHP script using a function to calculate the factorial of a number (non-negative integer). The function accepts the number as a argument.
19. Write a PHP script to get the first element of the below array. \$color = array(4 => 'white', 6 => 'green', 11=> 'red'); Output: white
20. Write a PHP script to calculate and display average temperature, five lowest temperatures.
21. Create a PHP script using a function to check a number is prime or not.
22. Create a PHP program which iterates the integers from 1 to 100. For multiples of three print "Fizz" instead of the number and for the multiples of five print "Buzz". For numbers which are multiples of both three and five print "FizzBuzz".
23. Create a web application form which includes nickname, first name, memo as field. Ask the user to enter the values and display the same on the web page.
24. Create a simple web application in HTML form and accept the user's name and display the name through PHP echo statement.
25. Write a simple PHP program to check that emails are valid.
26. Write a PHP script to count number of lines in a file.
27. Create a PHP script that demonstrate whether cookie is enabled or not.
28. Create a PHP script that has two session variable called color and animal. Set the value of both session variable and access the session information on another page and print it.
29. Create a PHP script that has one session variable called counter which increments its value as many times user visits the page.
30. Create a PHP script that will ask user to upload a image file in a folder and checks whether it is successfully uploaded or not.
31. Create a web page for simple Login and Logout using sessions.
32. Create a web page that will ask the user to input email address and give validation for proper email address.
33. Write a PHP script to connect the server and database.
34. Create a PHP web page that will ask the user to insert records to the table in Database.
35. Create a PHP web page that will fetch records from the table in Database.
36. Create a PHP web page that will upload image and read the image from the database and display on the web page.
37. Create a web form for Registration with required fields and ask user to input the data and display the same in the table.

38. Create a web form for storing car details. Take all the necessary fields like model no, price, company etc and ask the user to input the details of car and submit it to store into database.
39. Create a web page of feedback with required fields like username, comments etc and ask user to give the feedback and store it in the table in the database.
40. Create a web page that will display all the details of order received and allows user to update the order and store it in a table.
41. Create a Contact us web page with required fields like name, email_id, message etc and ask user to enter the details and store it in a database.
42. Create a web page that will ask user to select any one option whether user wants to eat punjabi, chinese, italian or mexican. Give choices using radio button, when user selects any cuisine display in the message box and store it into the table.

PO-COMPETENCY-CO MAPPING

CO \ PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	3	2	2	3	1				2		
CO2	3	3	2	3	1				2	1	
CO3	3	3	3	3	2	1	1		2	1	
CO4	2	2	3	3	1	1			2		
CO5	3	3	3	3	2	2	1		3	2	2

Course: Implementation Of Website Frameworks II								
Course Code:		Credit:4	Semester: V			Programme: Integrated B.Sc.-M.Sc. (CA & IT)		
Teaching Scheme				Evaluation Scheme				
Lecture	Tutorial	Practical	Contact Hours	CIE Theory	CIE Practical	ESE Theory	ESE Practical	Total
2	0	4	6	18/50	-	18/50	-	36/100

Course Outcomes (COs)

After completion of this course, students will be able to:

- **CO1:** Understand the architecture of the Django framework and its role in web application development.
- **CO2:** Apply Django ORM to design and manage relational databases effectively.
- **CO3:** Develop dynamic web applications with forms, authentication, and session handling.
- **CO4:** Integrate front-end resources (templates, CSS, JS, media files) within Django projects.
- **CO5:** Implement email services, messaging systems, and REST APIs to enhance application functionality.
- **CO6:** Deploy Django applications on local servers and cloud platforms.

Course Learning Outcomes (CLOs):

After successful completion of this course, a student will be able to:

1. **Explain** the structure of Django projects and apps, and **demonstrate** URL mapping and template rendering.
2. **Design** and **implement** database models using Django ORM, and **perform** CRUD operations via views and the Django shell.
3. **Develop** user authentication, login, and session management features by **applying** Django's built-in authentication system.
4. **Integrate** static/media files, **implement** email functionality, and **deploy** Django applications to a server or cloud platform.
5. **Utilize** Django REST Framework (DRF) to **build** simple RESTful APIs for web applications.

Module: I Getting Started with Django(25%)

What is Django?, Django project and app creation, Understanding the folder structure, reading views and mapping URLs, Using templates to display content

Module: II Models and Admin Panel (25%)

What is Django ORM? Creating models (tables), Using migrations to create database structure, Registering models in the admin panel, Basic CRUD operations using Django shell and views

Module: III Forms, Login System, and Sessions (25%)

Creating forms (manual and ModelForm), Validating form inputs, Creating login and registration pages, Using Django's built-in authentication, Managing user sessions and roles

Module: IV Static Files, Email, and Deployment (25%)

Using static files (CSS, JS) and media files (images, uploads), Sending emails from contact forms, Showing messages (success/error), Introduction to Django REST Framework (DRF), Deploying Django app on Heroku or local server

List of suggestive programs.

1. Create a Basic Django Project
Initialize a Django project and run the development server.
2. Create and Link an App
Create a Django app and map a view to the home page using urls.py.
3. Create Multiple Views and Route Them
Set up 3–4 basic views like "About", "Contact", and "Services", and link them using URLs.
4. Use Templates to Display Data
Pass variables from view to template and render dynamic content.
5. Create and Migrate a Model
Build a model like Student or Product, and apply migrations to create the table.
6. Register Model in Admin Panel
Make the model accessible in the Django admin and customize its display.
7. Perform CRUD Operations Using Django Shell
Create, read, update, and delete entries using the shell (python manage.py shell).

8. Create a Simple View for CRUD Display
9. Display all model entries using a view and template (Model.objects.all()).
10. Create a Manual HTML Form
Accept input (e.g., name, email) and show it back on another page.
11. Create a ModelForm for a Model
Use Django's ModelForm to create a form directly from a model.
12. Create a User Registration Page
Let users sign up and store credentials using Django's User model.
13. Add Login & Logout Functionality
Use Django's built-in authentication to log users in/out.
14. Set Up a Protected View
Show content only if a user is logged in (using @login_required).
15. Use Static and Media Files
Add a CSS file and an image to your template using Django's static system.
16. Send Email from Contact Form
Build a form that sends an email when submitted (can use console backend for testing).

Text Books:

1. **"Django for Beginners" by William S. Vincent**
– A practical, beginner-friendly guide to building web applications using Django step-by-step.
2. **"Mastering Django: Core" by Nigel George**
– Comprehensive coverage of Django fundamentals, project structure, views, models, and templates.

Reference Books:

1. **"Two Scoops of Django" by Audrey Roy Greenfeld & Daniel Roy - Greenfeld**
– Best practices and practical tips for writing clean and efficient Django code.
2. **"Web Development with Django Cookbook" by Jake Kronika and Aidas Bendoraitis**
– Problem-solving recipes for real-world Django development and deployment.

PO-COMPETENCY-CO MAPPING

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	3	2	1	1	2	1	-	1	-	2	-
CO2	3	3	2	2	3	2	-	2	-	2	-
CO3	3	3	3	3	3	2	-	2	2	3	-
CO4	2	2	3	3	3	2	-	2	2	2	-
CO5	2	2	3	3	3	3	-	3	2	3	-
CO6	2	2	3	3	3	3	-	3	2	3	-

Course: Software Engineering								
Course Code: BCA51CC03			Semester: V			Programme: BCA		
Teaching Scheme				Evaluation Scheme				
Lecture	Tutorial	Practical	Contact Hours	CIE Theory	CIE Practical	ESE Theory	ESE Practical	Total
2	0	0	2	18/50	-	18/50	-	36/100

Course Outcomes (COs)

After completion of this course, students will be able to:

- **CO1:** Explain the fundamentals of software, its types, characteristics, myths, and role of Software Engineering.
- **CO2:** Compare and apply various software process models, including traditional and agile approaches.
- **CO3:** Apply requirements engineering techniques and design principles to develop structured SRS and basic system design.
- **CO4:** Apply software testing methods and quality assurance principles to ensure reliable and standard-compliant software.

Course Learning Outcomes (CLOs):

After successful completion of this course, a student will be able to:

1. **Explain** the fundamental concepts of software engineering, including the nature of software, its types, characteristics, myths, and professional ethics.
2. **Compare** and **analyze** different software process models (traditional and agile) and **apply** suitable methodologies for software development.
3. **Apply** requirements engineering techniques to **capture, specify, and document** functional and non-functional requirements in the form of an SRS.
4. **Design** software solutions using architectural models, component-level design, and user-interface design principles.
5. **Apply** software testing strategies, **design** test cases using black-box and white-box techniques, and **evaluate** software quality using standards and reviews.

Course Content

Module I: Introduction to Software Engineering (25%)

Definition and Nature of Software, Characteristics of Good Software, Types of Software: System, Application, Embedded, Web.

Introduction to Software Engineering, Importance of Software Engineering in Modern Computing, Software Engineering Ethics and Professional Practice, Software Myths and Realities

Module: II Software Process Models (25%)

Software Process: Definition and Activities (Communication, Planning, Modeling, Construction, Deployment), Prescriptive Process Models: Linear Sequential (Waterfall), Prototyping, RAD, Evolutionary Models: Incremental, Spiral, Agile Software Development: Principles and Practices, Overview of Scrum and Extreme Programming (XP), Comparison of Traditional and Agile Models

Module III: Requirements and Software Design (25%)

Requirements Engineering: Importance and Activities, Functional vs. Non-Functional Requirements, Requirement Elicitation Techniques (Interviews, Questionnaires), Requirements Specification Document (SRS): Structure and Sample, Basic Design Principles and Concepts, Architectural Design: Layered and Client-Server Architecture, Component-Level Design and Simple UI Design Principles

Module IV: Software Testing and Quality Overview (25%)

Importance of Testing in SDLC, Testing Levels: Unit, Integration, System, Acceptance, Testing Techniques: White-box (statement & branch coverage), Black-box (equivalence partitioning, boundary value analysis), Basic Test Case Design, Introduction to Software Quality Assurance (SQA), Software Review Techniques (Walkthroughs, Inspections), Quality Standards Overview: ISO 9000, CMM Levels

Text Books:

1. Pressman Roger S., Software Engineering – A Practitioner's Approach, 8th Edition TATA McGraw Hill Publications.

Reference Books:

1. Sommerville, Software Engineering, 8th Edition Pearson Education.

2. Jawadekar Waman S., Software Engineering – Principles and Practices, TMGH Publication
3. Mall Rajib, Fundamentals of Software Engineering, Prentice-Hall, 2011.
4. Mishra Jibitesh and Mohanty Ashok, Software Engineering, PERSON
5. Datta Subhajit, Software Engineering Concept and Application, OXFORD
6. Jalote Pankaj, Software Engineering – A Precise Approach, Wiley India
7. Jawadekar Waman S., Software Engineering – A Primer, TMGH Publication
8. Pfleeger Shari Lawrence and Atlee Joanne M., Software Engineering – Theory and Practice, 3rd Edition, Pearson Education.

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



SHREYARTH UNIVERSITY

School of Computer Science and Applications

**Master of Science (Computer
Applications & Information Technology)**

Semester: I & VII

Program Outcomes

PO 1	Computational Knowledge
PO 2	Problem Analysis
PO 3	Design /Development of Solutions
PO 4	Conduct investigations of complex Computing problems
PO 5	Modern Tool Usage
PO 6	Professional Ethics
PO 7	Life-long Learning
PO 8	Project management and finance
PO 9	Communication Efficacy
PO 10	Societal and Environmental Concern
PO 11	Individual and Team Work
PO 12	Innovation and Entrepreneurship

Program Specific Outcomes

PSO 1	Understand and apply knowledge on analysis, design and development of applications in the computing discipline.
PSO 2	Use of recent technology, skill and knowledge for computing practice with commitment on societal, moral values.
PSO 3	Inculcate employability and entrepreneur skills among students who can develop customized enterprise level solutions.
PSO 4	Develop techniques to enhance ability for lifelong learning.

Course: Object Oriented Programming with Java								
Course Code:		Credit:4		Semester: I			Programme: M.Sc. (CA & IT)	
Teaching Scheme				Evaluation Scheme				
Lecture	Tutorial	Practical	Contact Hours	CIE Theory	CIE Practical	ESE Theory	ESE Practical	Total
3	0	2	5	25/50				

Course Objectives

- Understand and apply object-oriented programming concepts.
- Develop Java applications using classes, inheritance, interfaces, and exception handling.
- Use Java libraries for file handling, collections, GUI development, and multithreading.
- Build strong foundations for future enterprise and mobile Java applications.

Course Learning Outcomes (CLOs)

- By the end of this course, students will be able to:
- Understand and explain** the fundamental concepts of object-oriented programming such as classes, objects, inheritance, polymorphism, abstraction, and encapsulation using Java.
(Blooms Level: Understand)
- Apply** Java syntax and core programming constructs to develop basic to intermediate-level applications.
(Blooms Level: Apply)
- Design and implement** Java programs using advanced features like interfaces, packages, exception handling, and file I/O.
(Blooms Level: Apply)
- Utilize** Java Collection Framework and multithreading concepts to write efficient and concurrent programs.
(Blooms Level: Apply/Analyze)
- Develop** interactive desktop applications using AWT/Swing components with appropriate event handling mechanisms.
(Blooms Level: Create)
- Demonstrate** debugging and problem-solving skills through lab work and mini-projects in real-world Java applications.
(Blooms Level: Apply/Evaluate)
- Work effectively** in a team or individually to design, implement, and present object-oriented solutions for a given problem.
(Blooms Level: Evaluate/Create)

Module 1: OOP Concepts and Java Basics

Topics:

Introduction to Object-Oriented Programming
 Java Basics: JVM, JRE, JDK, IDE setup
 Java Syntax: variables, data types, operators
 Control Structures: if-else, loops, switch-case
 Classes and Objects
 Constructors and Constructor Overloading
 Static Members and the this Keyword

Module 2: Core OOP Principles – Inheritance, Polymorphism, Encapsulation

Topics:

Inheritance: Types, super keyword
Method Overriding and Dynamic Dispatch
Abstract Classes and Interfaces
Encapsulation and Access Modifiers
Method Overloading vs Overriding
Packages and Visibility

Module 3: Exception Handling, Strings, Arrays, and Collections

Topics:

Exception Handling: Types, try-catch-finally, throw/throws
Custom Exceptions
Strings: String, StringBuilder, string manipulation
Arrays: Single and Multidimensional
Wrapper Classes and Autoboxing
Java Collections: ArrayList, LinkedList, HashSet, HashMap

Module 4: File I/O, Multithreading, and GUI with AWT/Swing

Topics:

File Handling: File, FileReader, BufferedReader, FileWriter
Thread Lifecycle, Creation with Runnable and Thread
Thread Synchronization and Communication
GUI Programming: AWT and Swing Components
Layout Managers and Event Handling

Lab Activities:

1. Writing Java programs using control statements
2. Class and object creation
3. Constructor and static method demos
4. Programs on inheritance and interface implementation
5. Creating and using custom packages
6. Access modifier demonstration
7. Programs to handle exceptions and user-defined exceptions
8. String and array-based problems
9. Student record systems using collections
10. File-based read/write applications
11. Multithreading demos
12. Mini GUI applications (e.g., Calculator, Registration Form)

Text Books:

"Programming with Java" by E. Balagurusamy
"Java: The Complete Reference" by Herbert Schildt

Reference Books:

"Core Java Volume I – Fundamentals" by Cay S. Horstmann
"Head First Java" by Kathy Sierra and Bert Bates

PO-COMPETENCY-CO MAPPING

CLO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CLO1	H	H	M	L	M	L	L	L	L	M	L	L
CLO2	H	H	H	M	H	L	L	L	M	M	M	L
CLO3	H	H	H	M	H	M	L	L	M	M	L	M
CLO4	H	H	M	M	M	M	M	L	M	M	M	M

CLO	PSO1	PSO2	PSO3	PSO4
CLO1	H	M	L	M
CLO2	H	H	M	M
CLO3	H	H	M	H
CLO4	H	H	H	H

Course: Python Programming								
Course Code:			Semester: I			Programme: MSCIT		
Teaching Scheme				Evaluation Scheme				
Lecture	Tutorial	Practical	Contact Hours	CIE Theory	CIE Practical	ESE Theory	ESE Practical	Total
3	0	2	5					

Course Objectives:

1. Introduce fundamental programming concepts using Python and set up the programming environment.
2. Teach Python syntax, data types, operators, and control flow statements for problem solving.
3. Develop skills to manipulate strings, lists, dictionaries, tuples, and sets.
4. Enable writing modular and reusable code with functions and object-oriented programming.
5. Equip students to handle exceptions and perform file operations in Python programs.

Course Learning Outcomes (CLOs):

- CLO1: Demonstrate the ability to write Python programs using basic syntax, variables, data types, operators, and control structures like conditionals and loops.
- CLO2: Manipulate and process strings and lists effectively using built-in methods and operations in Python.
- CLO3: Create and utilize Python dictionaries, tuples, and sets for efficient data storage and retrieval.
- CLO4: Develop modular Python programs using functions, apply object-oriented programming principles, handle exceptions, and perform file input/output operations.

Course Content

Module I: Overview of Python, Programming languages and Control Structure(20%)

Software development, thrust area of Python, Installing Python Jupiter notebook. Identifiers, Keywords, statements and expressions, variables, Operators, Precedence and Associativity, Data types, Indentation, comments, reading input, print output, Type conversions, type () function and Is Operator, Dynamic and strongly typed language. If decision control flow, if-else decision control flow, if..elif..else decision control, Nested if statement, while loop, for loop, continue and break statements

Module II: Python Data Types and Structures with String and List (15%)

String: Creating and Storing Strings, Basic String Operations, Accessing Characters in String by Index Number, String Slicing and Joining, String Methods, Formatting Strings. Lists: Creating Lists, Basic List Operations, Indexing and Slicing in Lists, Built-In Functions Used on Lists, List Methods, the del Statement.

Module III: Python Data type and Structure with Dictionary, Set and Tuple (15%)

Dictionaries: Creating Dictionary, Accessing and Modifying key, value Pairs in Dictionaries, Built In Functions used on Dictionaries, Dictionary Methods, The del Statement. Tuples and Sets: Creating Tuples, Basic Tuple Operations, Indexing and Slicing in Tuples, Built-In Functions Used on Tuples, Relation between Tuples and Lists, Relation between Tuples and Dictionaries, Tuple.

Module IV: Functions and Modular Programming with file Handling

Functions: Built-in functions, commonly used modules, definition and calling the function, return statement and void function, scope and lifetime of variables, Default Parameters, Keyword Arguments, Command Line Arguments. Classes and Objects: Inheritance, Encapsulation, Polymorphism, Constructor (`__init__`) and self, Class and Static Methods, Exception Handling. Files Handling: Types of Files, Creating and Reading Text Data, File Methods to Read and Write Data, Reading and Writing Binary Files.

Practical Lab List: Python Programming.

1. A simple Python program that uses the `print()` function to display a welcome message, along with your **name**, **course**, and **date of birth**.
2. Create a user-defined function to compute area of a circle.
3. Print N Fibonacci numbers recursively.
4. Print a numeric pyramid pattern using nested loops.
5. Write functions for basic calculator operations.
6. Function to check if a string is palindrome.
7. Use string built-in functions like `replace()`, `split()`, `join()`.
8. Return multiple values from a function (min, max, avg).
9. Use built-in math functions (`sqrt`, `pow`, `ceil`).
10. Implement a simple contact book using dictionaries.
11. Create and manipulate a list of numbers
12. Perform set operations: union, intersection, difference.
13. Nested dictionary to store multiple student records.
14. Define a Student class with attributes and display method.
15. Define a Student class with attributes and display method.
16. Implement inheritance with Person and Employee classes.
17. Handle exceptions in division by zero scenario using try-except-finally.
18. Count the number of lines, words in a text file.
19. Serialize and deserialize objects using pickle module.
20. Create a simple marksheet Program using class, Functions, Operators, Dictionary.

Text Books:

- Introduction to Python Programming”
- Publisher: Gowrishankar S, Veena

Reference Books: .

1. “Programming in Python 3: A Complete Introduction to the Python Language”, *Publisher: Mark Summerfield*
widely used as a comprehensive guide for mastering Python 3.
2. “Modular Programming with Python: introducing modular techniques for building sophisticated programs using Python”

Publisher: Erik Westra

structured and modular programming practices in Python, helping learners design scalable and maintainable software.

Reference Links:

- <https://realpython.com/python-basics/>
- <https://docs.python.org/3/tutorial/controlflow.html>
- https://www.w3schools.com/python/python_conditions.asp

Course: Data Structure								
Course Code:				Semester: I		Programme: MSC(CA & IT)		
Teaching Scheme				Evaluation Scheme				
Lecture	Tutorial	Practical	Contact Hours	CIE Theory	CIE Practical	ESE Theory	ESE Practical	Total
3	1	4	8	60	60	40	40	200

Course Outcomes: Name of CO Description

CO1 Understand and implement linear data structures such as arrays, stacks, queues, and linked lists.

CO2 Apply memory management techniques and convert and evaluate arithmetic expressions.

CO3 Analyze and perform operations on various types of trees and tree-based representations.

CO4 Apply graph traversal algorithms and implement efficient searching, sorting, and hashing techniques.

Course Learning Outcomes:

- Understand and implement fundamental linear data structures like arrays, stacks, queues, and linked lists.
- Apply concepts of dynamic memory management, expression conversion, and evaluation using stacks.
- Design and operate on various tree structures including binary trees, AVL trees, and B+ trees.
- Analyze and implement graph traversal techniques and solve problems using BFS, DFS, and shortest path algorithms.
- Evaluate and apply different sorting, searching, and hashing techniques for efficient data access and manipulation.

Unit - 1: Introduction to Data Structures and Linear Structures (25 %)

Introduction to Data Structures, Storage structure for arrays, Sparse matrices, Stacks and Queues: Representation and applications, Linked Lists: Singly Linked Lists, Linked list representation of Stacks and Queues, Operations on Polynomials, Doubly Linked List, Circular Linked List

Unit - 2: Expression Handling and Memory Management (20%)

Dynamic Storage Management, Garbage Collection and Compaction, Infix to Postfix Conversion, Postfix Expression Evaluation

Unit - 3: Non-Linear Data Structures – Trees (30%)

Tree Terminology, Binary Tree, Binary Search Tree (BST), General Tree, B+ Tree, AVL Tree, Complete Binary Tree, Tree Representation, Tree Traversals (Preorder, Inorder, Postorder), Operations on Binary Tree – Expression Manipulation

Unit - 4 : Graphs and Searching/Sorting Techniques (25%)

Graph Terminology, Representation of Graphs, Path Matrix, Breadth First Search (BFS), Depth First Search (DFS), Topological Sorting, Warshall's Algorithm (Shortest Path), Sorting Techniques: Bubble Sort, Selection Sort, Insertion Sort, Quick Sort, Merge Sort, Heap Sort, Radix Sort, Searching Techniques: Linear Search, Binary Search, Hashing Techniques and Hash Functions

Text Books:

1. "Data structure in C" by Tanenbaum, PHI publication / Pearson publication.

Page 3 of 5

2. Pai: "Data Structures & Algorithms; Concepts, Techniques & Algorithms" Tata McGraw Hill.

Reference Books:

1. "Fundamentals of data structure in C" Horowitz, Sahani & Freed, Computer Science Press.
2. "Fundamental of Data Structure" (Schaums Series) Tata-McGraw-Hill.

Practical List:

1. Implement array operations: insertion, deletion, searching.
2. Represent sparse matrices using array and perform addition.
3. Implement stack using arrays and linked lists.
4. Implement queue and circular queue using arrays.
5. Implement singly linked list with basic operations (insert, delete, traverse).
6. Implement linked list-based stack and queue.
7. Polynomial operations using linked lists (addition, multiplication).
8. Implement doubly and circular linked list with insert and delete operations.
9. Create a binary tree and perform inorder, preorder, and postorder traversals.
10. Implement operations on binary search tree (insert, delete, search).
11. Represent general tree and perform basic traversals.
12. Expression tree creation and evaluation.
13. Implement B+ tree insertion (basic simulation).
14. Represent complete binary tree using array and linked list.
15. Represent a graph using adjacency list and adjacency matrix.
16. Implement BFS and DFS traversal algorithms.
17. Perform topological sort on a directed acyclic graph.
18. Implement sorting algorithms: bubble, insertion, selection, quick, merge, heap, radix.
19. Implement linear and binary search.
20. Implement hashing with linear probing and chaining.

List of Open-Source Software/learning website:

1. <https://www.tutorialspoint.com>
2. <https://www.w3schools.com>
3. <https://www.javatpoint.com>

Course Outcome (CO)	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO1	L	H	H	H	L	L	M	L	L	L	L	L
CO2	M	M	L	L	L	L	L	L	L	L	L	L
CO3	H	H	H	M	M	L	M	L	M	L	M	M
CO4	H	M	H	M	M	L	M	L	M	L	M	M
CO5	H	H	H	M	M	L	M	L	M	L	M	M

Course Outcome (CO)		PSO 2	PSO 3	PSO 4
CO1	M	L	L	L
CO2	H	M	L	L
CO3	H	L	L	L
CO4	H	L	L	L
CO5	H	L	M	L

Course: Database Management System and NO SQL								
Course Code:			Semester: I			Programme: MSC- IT		
Teaching Scheme				Evaluation Scheme				
Lecture	Tutorial	Practical	Contact Hours	CIE Theory	CIE Practical	ESE Theory	ESE Practical	Total

Course Objectives:

- Understand the need for databases, data models, and DBMS architecture over traditional file systems.
- Design relational schemas using E-R modeling and apply normalization for efficient database design.
- Develop SQL skills for data definition, manipulation, and implement transaction management principles.
- Explore data storage techniques, indexing, and gain basic knowledge of NoSQL and its comparison with relational databases.

Course Learning Outcomes (CLOs):

By the end of the course, students will be able to:

1. Explain the need for database systems, data models, and DBMS architecture compared to traditional file systems.
2. Design relational database schemas using E-R modeling and normalization techniques for efficient data organization.
3. Apply SQL for data definition and manipulation, including advanced queries and transaction management concepts.
4. Analyze data storage methods and compare NoSQL and relational databases in terms of structure, performance, and scalability.

Course Content

Module 1: Relational Database Concepts & SQL Review

Introduction to DBMS and RDBMS, Data Models (Relational, Hierarchical, Network), Entity-Relationship (ER) Model and ER Diagrams, Relational Algebra and Calculus (Basics), SQL Review: DDL, DML, DCL, TCL, Joins, Subqueries, Views, Indexing, Stored Procedures, Triggers

Module 2: Database Design and Normalization

Functional Dependencies, Normal Forms (1NF to BCNF, 4NF, 5NF), Decomposition and Lossless Join, Dependency Preservation, Schema Refinement, Design Using ER-to-Relational Mapping

Module 3: Transaction Management and Query Optimization

Transactions and ACID Properties, Concurrency Control (Locking protocols, Deadlock handling) Recovery Techniques, Database Security and Authorization, Query Processing and Optimization

Module 4: Introduction to NoSQL Databases and Applications

Evolution of NoSQL, Differences between SQL and NoSQL, CAP Theorem (Consistency, Availability, Partition Tolerance), Types of NoSQL Databases: Document-Oriented, Column-Oriented, Key-Value Stores, Graph-Based, MongoDB: Installation and Configuration, CRUD Operations, Aggregation, Indexing, Data Modeling, Cassandra: Architecture and CQL (Cassandra Query Language), Neo4j: Introduction to Graph Databases and Cypher Queries, Use Cases and Applications of NoSQL in Big Data, IoT, Social Media, etc.

Text Books:

1. **Database System Concepts** by Silberschatz, Korth, and Sudarshan
2. **NoSQL Distilled** by Pramod J. Sadalage & Martin Fowler

References Books:

1. **MongoDB: The Definitive Guide** by Kristina Chodorow
2. **Cassandra: The Definitive Guide** by Eben Hewitt
3. **Learning Neo4j** by Rik Van Bruggen

List of suggestive programs:

Program 1: Define a University database with tables like Students, Courses, and Enrollments using SQL DDL.

Program 2: Insert sample records, update student names, and delete a course record using DML commands.

Program 3: Write SQL queries using INNER JOIN, LEFT JOIN and nested subqueries to fetch student-course details.

Program 4: Create a view to display enrolled students and use indexing on student roll numbers for faster retrieval.

Program 5: Write a trigger that logs deletions in a log table and a stored procedure to assign grades based on marks.

Program 6: Design an ER diagram for a Library Management System and map it into relational tables.

Program 7: Given a relation schema with anomalies, identify FDs and normalize from 1NF → 2NF → 3NF.

Program 8: Prove lossless decomposition and dependency preservation for a sample relation.

Program 9: Take a relation and decompose it to BCNF and further to 4NF with explanation.

Program 10: Given a business case (e.g., Hospital), refine the schema using normalization and ER-to-relational mapping.

Program 11: Use SQL to implement transactions and illustrate atomicity, consistency, isolation, and durability.

Program 12: Simulate 2 transactions with shared/exclusive locks and show possible deadlock or serializability.

Program 13: Use EXPLAIN or QUERY PLAN to analyze query cost and suggest improvements (e.g., using indexes).

Program 14: Write transaction control code to use SAVEPOINT, ROLLBACK TO, and COMMIT.

Program 15: Grant/revoke privileges using DCL commands and demonstrate different user access levels.

Program 16: Create a collection products and perform Create, Read, Update, Delete operations.

Program 17: Perform group-by operations using aggregation pipeline and create an index on a field.

Program 18: Install Cassandra, create a keyspace, tables, and perform basic CQL queries.

Program 19: Create a simple social network graph and write Cypher queries to find friends-of-friends.

Program 20: Store a sample dataset (e.g., user comments) in both MySQL and MongoDB, and compare performance of similar queries.

Program 1: Write a comparison report (or simple CLI tool) showing differences between File System and DBMS through examples.

Program 2: Create a basic DBMS user role chart showing different user types (admin, end-user, Program 3: Create a table structure in MySQL/PostgreSQL for a Student Management System including primary, foreign, and candidate keys.

Program 4: Write a program/script to enforce domain and referential integrity using SQL CHECK, Program 5: Draw an E-R diagram for a Library Management System using any diagram tool (draw.io, Lucidchart) or manually.

Program 6: Convert your E-R diagram to SQL table creation scripts (map entities to tables and relationships to foreign keys).

Program 7: Create a database for a Hospital Management System and perform:

- INSERT, UPDATE, DELETE, and SELECT operations.
 - Apply WHERE, ORDER BY, and GROUP BY clauses.
- Program 8: Write queries using:

- JOINS (inner, outer, self joins)
- SUBQUERIES
- VIEWS
- DDL: CREATE, ALTER, DROP
- DML: INSERT, UPDATE, DELETE

Program 9: Given a raw unnormalized table, apply:

- 1NF (remove repeating groups)
- 2NF (remove partial dependencies)
- 3

N
F
/
B
C
N
F
(r
e
m
o
v
e
t
r
a
n
s
i
t
i
v
e
d
e
p
e
n
d
e
n
c
i
e
s
)
I
m
p
l
e

m
e
n
t
u
s
i
n
g
S
Q
L
t
a
b
l
e
t
r
a
n
s
a
c
t
i
o
n
s.

Program 10: Write a SQL program to simulate two transactions:

- Use START TRANSACTION, COMMIT, ROLLBACK
- Demonstrate ACID properties and consistency

Program 11: Simulate concurrency issues like lost updates or dirty reads using multiple client connections (with InnoDB engine).

Program 12: Write a program/script to:

- Analyze how MySQL stores tables using SHOW TABLE STATUS;
- Demonstrate single-level vs. multi-level indexing (conceptually or using sample indexes)

Program 13: Simulate hashing using a programming language (e.g., Python or Java) to show how keys map to data blocks.

Program 14: Using MongoDB or any NoSQL tool:

- Create a key-value store for a product inventory

- Insert and retrieve documents

Program 15: Compare MongoDB collections to SQL tables by implementing the same data structure in both and observing differences.

PO-COMPETENCY-CLO MAPPING:

For CLO-PO mapping:

CL Os	P O 1	P O 2	P O 3	P O 4	P O 5	P O 6	P O 7	P O 8	P O 9	PO 10	PO 11	PO 12
CL O1	M	H	M	M	L	H	M	H	M	L	M	M
CL O2	H	M	H	H	H	M	H	L	H	M	H	H
CL O3	M	H	M	H	M	L	M	H	M	H	M	M
CL O4	M	H	H	H	H	M	M	L	H	M	M	H

For CLO-PSO mapping:

CLO	PSO1	PSO2	PSO3	PSO4
CLO1	M	H	M	M
CLO2	H	L	M	H
CLO3	M	L	H	M
CLO4	M	H	H	M

Course: Agile Methodology and Development Practices									
Course Code:						Semester:		Programme: MSc.IT	
Teaching Scheme						Evaluation Scheme			
Lecture		Tutor ial	Practi cal	Cont act Hour s	CIE Theo ry	CIE Practi cal	ESE Theo ry	ESE Practi cal	Tot al
2		0	0	2		-		-	

Course Objectives

Having completed this course, the learner will be able to

1. Understand Agile values, principles, and lifecycle.
2. Differentiate Agile from traditional models.
3. Explore key Agile methodologies.
4. Study Git as a modern version control system used in Agile teams
5. Learn foundational software development practices aligned with Agile.
6. Understand how Agile and DevOps align in modern development workflows

Course Learning Outcomes (CLOs):

1. Describe the core values, principles, and lifecycle of Agile software development.
2. Compare Agile methodologies with traditional software development models such as Waterfall, highlighting key differences in process and mindset.
3. Explain the features, roles, and workflows of major Agile methodologies including Scrum, Kanban, and Extreme Programming (XP).
4. Illustrate the role of Git in Agile environments and explain the conceptual use of version control for collaboration and code management.
5. Discuss essential software development practices such as Test-Driven Development (TDD), Continuous Integration/Delivery (CI/CD), Refactoring, and Code Review.
6. Analyze the integration of Agile with DevOps culture and its impact on modern software delivery processes.

Course Content

Module: I: Agile Software Development – Principles & Frameworks(25%)

The Agile Manifesto: Values and Principles, Comparison with traditional models (Waterfall, Spiral), Agile vs Iterative vs Incremental, Overview of Agile methodologies: Scrum (roles, events, artifacts), Kanban (boards, WIP limits), Extreme Programming (XP) basics, Agile team structure and communication, Agile lifecycle.

Module: II Scrum & Agile Estimation Techniques (25%)

Scrum roles: Product Owner, Scrum Master, Development Team, Scrum events: Sprint, Daily Stand-up, Sprint Planning, Review & Retrospective, Scrum artifacts: Product Backlog, Sprint Backlog, Increment, Agile estimation techniques: Story points, Planning poker, Velocity and burn-down chart, Agile metrics and success factors, Agile myths and common challenges.

Module: III Git & Version Control Concepts (25%)

Basics of Version Control: Centralized vs Distributed Version Control Systems, Introduction to Git: Key concepts: repository, commit, branch, merge, Git lifecycle: init, add, commit, push, pull, Benefits of Git in Agile teams, Branching strategies: Feature branches, GitFlow overview, Collaborative development using Git.

Module: IV Development Practices in Agile Projects (25%)

Introduction to modern development practices: Test-Driven Development (TDD), Refactoring, Code reviews (peer review process), Pair programming, Introduction to Continuous Integration & Continuous Delivery (CI/CD), Agile Testing: unit, integration, acceptance testing concepts, Introduction to DevOps—Overview, DevOps - Importance and Benefits, DevOps Principles and Practices, 7-C's of DevOps Lifecycle for Business Agility.

Text Books:

Robert C. Martin, "Agile Software Development, Principles, Patterns and Practices", First International Edition, Prentice Hall.

Reference Books:

1. David J. Anderson and Eli Schragenheim, —Agile Management for Software Engineering: Applying the Theory of Constraints for Business ResultsII, Prentice Hall, 2003.
2. Hazza and Dubinsky, —Agile Software Engineering, Series: Undergraduate Topics in Computer ScienceII, Springer, 2009.
3. Craig Larman, —Agile and Iterative Development: A Manager's Guide, Addison-Wesley, 2004.
4. Kevin C. Desouza, —Agile Information Systems: Conceptualization, Construction, and ManagementII, Butterworth-Heinemann, 2007.
5. Andrew Stellman, Jennifer Greene - Learning Agile: Understanding Scrum, XP, Lean, and Kanban, O Reilly, 2015.

Course: Communication Skills								
CourseCode:			Semester:			Programme:		
Teaching Scheme				EvaluationScheme				
Lecture	Tutorial	Practical	Contact Hours	CIE Theory	CIE Practical	ESE Theory	ESE Practical	Total

**MODULE
1**

Unit 1:

Fundamentals of Technical Communication

(7-10 lectures)

- Technical Communication: Features;
 - Distinction between General and Technical Communication;
 - Language as a tool of Communication;
 - Dimensions of Communication: Reading & comprehension;
 - Technical writing: sentences; Paragraph; Technical style: Definition, types & Methods;
 - Communication networks: on the basis of system
1. On direction: Up- ward, Down- ward, Horizontal, Vertical
 2. On Channel: Y network, Informal communication, Wheel network, Chain network, Circle network, All channel networks

- Barriers to Communication.

Unit 2: Forms of Technical Communication (Writing based)

(7-10 lectures)

- Technical Report: Definition & importance;
- Thesis/Project writing: structure & importance;
- synopsis writing: Methods;
- Technical research Paper writing: Methods & style;
- 7 Cs of effective business writing;
- C.V./Resume writing;
- Technical Proposal: Types, Structure & Draft.
- Ethics of online and offline meetings

MODULE 2

Unit 3: Technical Presentation:

(7-10 lectures)

- Strategies & Techniques Presentation:
- Forms; interpersonal Communication;
- Class room presentation; style; method;
- Individual conferencing: essentials: Public Speaking: method;
- Techniques: Clarity of substance; emotion; Humour;
- Modes and etiquette- Online offline meeting /Presentation;
- Overcoming Stage Fear
- Audience Analysis & retention of audience interest;
- Methods of Presentation:
- Interpersonal; Impersonal;

Unit 4: Technical Communication Skills Interview skills;

(3-4 lectures)

- Group Discussion: Objective & Method;
- Seminar/Conferences Presentation skills: Focus;
- Content; Style; Argumentation skills: Devices: Analysis;
- Cohesion & Emphasis;
- Critical thinking;
- Nuances: Exposition narration & Description;
- effective business communication competence: Grammatical;
- Discourse competence: combination of expression & conclusion; Socio-linguistic competence: Strategic competence: Solution of communication problems with verbal and non-verbal means.

Unit 5:

- Body language (National and International differences in Body language) **(2-3 lectures)**



SHREYARTH UNIVERSITY

**School of Computer Science and
Applications**

**B.Sc. in Cyber Security, Networking &
Cloud Computing (Hons.)**

SYLLABUS

SEMESTER - III

SHREYARTH UNIVERSITY, AHMEDABAD

SEMESTER-III Teaching Scheme W.E.F. YEAR 2025-26

Sr. No.	Sub. Code	Name of the subject	Curricular Type	CREDIT	Teaching Scheme				Examination Scheme		
					Th.	Tut.	Pr.	Total (Hr.)	CCE	SEE	Total marks
1		Cyber Security -I	Major Course-1	4	3	0	1	5	50	50	100
2.		Advance Python Programming	Major Course-2	4	3	0	1	5	50	50	100
3.		Introduction to JavaScript	Minor Course-1	4	3	0	1	5	50	50	100
4.		Information Security	MDC-1	4	3	0	1	5	50	50	100
5.		Micro Controller & Processor Programming	AEC	4	2	1	1	3	50	50	100
6.		Communication Skills III	SEC	2	2	0	0	2	25	25	50
7.		Indian Knowledge System II	IKS-II	2	2	0	0	2	25	25	50
		TOTAL		22	18	1	8	25	300	300	600

Cyber Security - I								
Course Code: CYM201-2C				Semester:3			Programme: B. Sc.in Cyber Security, Networking & Cloud Computing (Hons.)	
Teaching Scheme				Evaluation Scheme				
Lecture	Tutorial	Practical	Contact Hours	CIE Theory	CIE Practical	ESE Theory	ESE Practical	Total
3	0	1	5	18/50	-	18/50	-	36/100

Course Objectives:

- Identify and distinguish between the different types of Cybercrimes
- Identify the method of performing various cyber-attacks using different malwares
- Classify different types of digital forensic methods and IT laws by the use of case studies
- To assess and evaluate the computer networks and ports using network scanning tools

Course Learning Outcomes (CLOs):

1. **Understand Core Concepts.** Define essential cyber security terms and concepts, including confidentiality, integrity, and availability.
2. **Identify Common Threats.** Recognize and categorize various cyber threats, such as viruses, phishing attacks, and social engineering tactics.
3. **Implement Basic Security Practices.** Apply fundamental security measures, including password management, software updates, and safe browsing habits.
4. **Recognize the Importance of Cyber Hygiene.** Explain the significance of maintaining good cyber hygiene to protect personal and organizational data.

Course Content:

Module 1: Introduction to Cyber Crime:

Definition and Origin of the Word, Cyber Crime and Information Security, Cyberspace, Cyber Security: Definition, Who are Cyber Criminals, Classification of Cyber crimes, Basic Terminologies: Vulnerability, Threat, Exploit, Attack, Active Attacks, Passive Attacks, Types of hackers, How Criminal Plans the Attack, What is hacking , Phases of Hacking.

Module 2: Introduction to Different Cyber Attacks:

What is malware, Types of malwares: Virus, Worms Trojan, backdoors, Keyloggers and Spyware, Proxy server and Anonymizers, Dos and Ddos Attacks, SQL Injection, Buffer Overflow, E-mail Spoofing, Cyber Defamation, Salami Attack, Data Diddling, Forgery, Online Fruads, Software Piracy Computer Sabotage, Email Bombing, Computer Network Intrusion, Password Sniffing, Credit Card Frauds, Identity Theft, Social Engineering, Botnets, Phishing

Module 3: Understanding Digital Forensics Cyber Law and Vulnerability Scanning :

Introduction to Incident Response, Digital Forensics, Need for Computer Forensic, Digital Forensic Life Cycle, The Indian IT ACT 2000, Overview of vulnerability scanning, Vulnerability Examples, Network Reconnaissance – Nmap, Networks Vulnerability Scanning - Netcat, Network Sniffers and Injection tools – Tcpcat and Wireshark, Web Application Scanner: Vega, DVWA, Password Cracking and Brute Force Tools – John the Ripper

Module 4: Introduction to Network Defense:

Firewall Basics, Packet Filter Vs Firewall, How a Firewall Protects a Network, Stateless Vs Stateful Firewalls, Network Address Translation (NAT), Open Port, Port Forwarding, Snort: Intrusion Detection System

Textbooks:

1. Introduction to Programming Using Python by Y. Daniel Liang, Pearson
2. Python Programming: Using Problem Solving Approach by Reema Thareja, Oxford University Press
3. Python the Complete Reference by Martin C Brown, Tata Mcgraw Hill, India

Reference Books:

1. Programming and Problem Solving with Python by Ashok Kamthane and Amit Ashok Kamthane, Tata McGraw Hill, India
2. John V Guttag. "Introduction to Computation and Programming Using Python", Prentice Hall of India

List of Open-Source Software/learning website:

1. www.nptel.ac.in
2. <https://docs.python.org/3/tutorial/>
3. <https://www.w3schools.com/python/>

Advanced Python Programming								
Course Code: CYM202-2C				Semester:3		Programme: B.Sc.in Cyber Security, Networking & Cloud Computing (Hons.)		
Teaching Scheme				Evaluation Scheme				
Lecture	Tutorial	Practical	Contact Hours	CIE Theory	CIE Practical	ESE Theory	ESE Practical	Total
3	0	1	5	18/50	-	18/50	-	36/100

Course Objectives:

- Introduce students to Python with a focus on cyber security use cases.
- Enable students to automate tasks like scanning, enumeration, packet analysis, and password cracking.
- Help students develop basic penetration testing tools using Python.
- Build an understanding of ethical scripting and cyber forensics through hands-on practice.

Course Learning Outcomes (CLOs):

1. **Master Object-Oriented Programming (OOP).** Implement advanced OOP concepts such as inheritance, polymorphism, and encapsulation in Python.
2. **Utilize Python Libraries and Frameworks.** Leverage popular Python libraries and frameworks (e.g., NumPy, Pandas, Flask) to develop robust applications and perform data analysis.
3. **Implement Asynchronous Programming.** Understand and apply asynchronous programming techniques using asyncio to improve application performance and responsiveness.
4. **Develop and Manage APIs.** Create and manage RESTful APIs using Python, enabling seamless communication between different software applications.

Course Content:

Module 1: Functions, Modules, and Exception Handling:

Function declaration and definition, recursion, docstrings, scope and lifetime, return statement
Types of arguments (*args, **kwargs)
Generating random numbers, Math module
Creating and importing modules, packages
Exception handling (try, except, finally, custom exceptions)

Module 2: File Handling & Python Libraries:

Reading/writing text and binary files
File metadata, directory structure (os, shutil)
pip, venv, installing and using third-party libraries
Standard libraries: os, sys, datetime, math, json
Introduction to useful libraries: requests, socket, hashlib

Module 3: Python for Cyber Security Essentials:

Hashing with hashlib (MD5, SHA1, SHA256) Hashing passwords and files, Detecting tampering by comparing hashes, Simulating safe dictionary-based password guessing, Accessing public APIs for cybersecurity data (IP geolocation, threat feeds)

Module 4: File, Log Analysis and Basic Network Scanner:

Reading system and application log files using Python Searching for keywords and suspicious patterns (e.g., failed login attempts), Extracting IPs and timestamps from logs Writing summary reports based on parsed log data, Creating a basic port scanner for localhost/test systems Checking for open ports and common service banners, Understanding real-world implications of exposed ports (Optional: safe demo of scanning with nmap output parsing)

Textbooks:

1. Introduction to Programming Using Python by Y. Daniel Liang, Pearson
2. Python Programming: Using Problem Solving Approach by Reema Thareja, Oxford University Press
3. Python the Complete Reference by Martin C Brown, Tata Mcgraw Hill, India

Reference Books:

1. Programming and Problem Solving with Python by Ashok Kamthane and Amit Ashok Kamthane, Tata McGraw Hill, India
2. John V Guttag. "Introduction to Computation and Programming Using Python", Prentice Hall of India

List of Open-Source Software/learning website:

1. www.nptel.ac.in
2. <https://docs.python.org/3/tutorial/>
3. <https://www.w3schools.com/python/>

List of suggestive programs:

1. Write Python functions to calculate factorial (with recursion and without).
2. Use `*args` and `**kwargs` in a function to print arbitrary arguments.
3. Generate random numbers for a password generator using the random and string modules.
4. Create custom math module with functions like prime checker, power, etc., and import it.
5. Raise and handle exceptions using try-except-finally (e.g., divide by zero, invalid inputs).
6. Create a package containing modules for math and string operations.
7. Write and read data from text and binary files (e.g., store login records).
8. Traverse directories and fetch file details using `os`, `os.path`, and `shutil`.
9. Use `json` module to read/write structured data to a `.json` file.
10. Create and activate virtual environment, install third-party libraries using `pip`.
11. Use `datetime` and `sys` modules to log system information or generate timestamps.
12. Build a simple TCP client-server program using `socket`.
13. Extract IP address and hostname of your machine.

14. Send a web request using `requests` to fetch a website and extract status code, headers.
15. Scrape a basic HTML page using `BeautifulSoup` and extract email or links.
16. Write a Python script to extract URLs/emails from a text using `re` (OSINT mini-task).
17. Hash a string and file using `hashlib` (MD5, SHA1, SHA256).
18. Verify integrity of two files by comparing hashes.
19. Simulate dictionary-based password cracking (test against hashed password).
20. Use public APIs (like `ipinfo.io` or `ipapi`) to get IP geolocation via Python.
21. Read system/application logs (e.g., Apache logs or sample `auth.log`).
22. Extract specific log entries containing errors or failed logins.
23. Search for suspicious IPs or patterns in logs using `regex`.
24. Generate a summary report of how many times certain events occurred.
25. Write a basic port scanner using `socket` to scan `localhost`.
26. Check open ports and try banner grabbing from HTTP, FTP, or SSH (test environment only).
27. (Optional) Parse `nmap` XML/greppable output and summarize open services.

Introduction to JavaScript								
Course Code: CYM203-2C				Semester: 3		Programme: B.Sc.in Cyber Security, Networking & Cloud Computing (Hons.)		
Teaching Scheme				Evaluation Scheme				
Lecture	Tutorial	Practical	Contact Hours	CIE Theory	CIE Practical	ESE Theory	ESE Practical	Total
3	0	1	5	18/50	-	18/50	-	36/100

Course Objectives

- To introduce students to the foundational and intermediate concepts of JavaScript for developing interactive web applications.
- To develop practical skills in manipulating the Document Object Model (DOM) and performing dynamic updates to web pages.
- To enable students to handle and process data using arrays, objects, and higher-order functions like map, filter, and reduce.
- To train students in implementing client-side form validation and writing modular, clean, and maintainable code.

Course Learning Outcomes (CLOs)

Upon completing this course, students will be able to:

1. **Understand and apply** JavaScript fundamentals including variables, data types, operators, control structures, and functions using different syntaxes.
2. **Work with strings, numbers, arrays, and objects** using built-in and advanced methods to manipulate and process data effectively.
3. **Implement functional programming techniques** such as map(), filter(), and reduce() for clean and efficient code logic.
4. **Create interactive web applications** by manipulating the DOM, validating forms, and dynamically updating page content and styles.

Course Content:

Module 1: JavaScript Fundamentals, Higher Order Functions:

Variables (var, let, const), Data Types (Primitive & Reference), Operators (Arithmetic, Comparison, Logical, etc.), Popup Boxes: alert(), prompt(), confirm(), Introduction to Functional Programming, map(), filter(), reduce()

Module 2: Control Flow, Loops and DOM Manipulation:

Conditional Statements: if, else, else if, Loops: for, while, do...while, Switch Case, Jump Statements: break, continue, Introduction to DOM, Selecting Elements (getElementById, querySelector, etc.), Creating and Inserting Elements, Modifying Content (innerText, innerHTML), Removing and Replacing Elements, Reading, Writing & Removing Attributes, DOM Traversal (parent, child, sibling nodes), Manipulating Styles and Classes

Module 3: Functions ,Scope,Event Handling & Form Validation:

Function Declaration & Expressions, Arrow Functions, Scope: Block, Function & Global Scope, Return Statements & Parameters, Event Handling & Event Listeners, Inline and external event handling, Using `addEventListener()`, Common events: `click`, `dblclick`, `mouseover`, `mouseout`, `keydown`, `keyup`, `submit`, `focus`, `blur`, `change`, Form Validation:, Accessing Form Elements via JavaScript, Validating Input Fields, Displaying Error Messages Dynamically

Module 4: Working with Data:

String Methods, Number Methods, Math Object Methods, Arrays and Their Methods (`push`, `pop`, `shift`, `unshift`, `splice`, `slice`, `indexOf`, `includes`, `join`, `sort`, etc.), Objects: Object Literals, Constructor Functions, Classes, Nested Objects

Textbooks:

1. BayrossIvan,WebEnabledCommercialApplicationsDevelopmentusingHTML,DHTML,Javascript,PerlCGI.

Reference Books:

1. Thomas A., Schneider Powell and Fritz, Java Script – The Complete Reference - SecondEdition,TATAMcGRAWHILLPublication
2. WiltonP,BeginningJavaScript,2ndEdition,WileyDreamTech,2004.

List of suggestive programs:

1. Write a script to demonstrate usage of variables, data types, and basic arithmetic operations.
2. Create a program using `prompt()` to take user input, and use `alert()` or `confirm()` to display messages.
3. Develop a program to find the greatest of three numbers using `if-else`.
4. Write a program to print the multiplication table of a number using `for` loop.
5. Create a program using `while` or `do...while` to calculate the factorial of a number.
6. Demonstrate `break` and `continue` using a loop that skips even numbers and stops at a certain value.
7. Create and invoke a normal function, function expression, and arrow function that performs a calculation.
8. Write a function that accepts a string and returns the number of vowels in it.
9. Demonstrate usage of string methods (`toUpperCase()`, `slice()`, `indexOf()`, etc.) in a small app.
10. Perform mathematical operations using the `Math` object (e.g., random number generator, power, round, etc.).
11. Create and manipulate arrays using methods like `push()`, `pop()`, `splice()`, `sort()`, etc.
12. Develop a program using object literals and nested objects to store and retrieve student data.
13. Create a constructor function or use a `class` to model a product with properties and methods.
14. Apply `map()` to double the values in an array.

15. Use `filter()` to extract odd numbers from an array.
16. Use `reduce()` to calculate the sum of all elements in an array.
17. Write a script to select and modify content of HTML elements (using `innerText`, `innerHTML`).
18. Create and insert new elements dynamically using JavaScript.
19. Write a program to change styles (like color, background) using JavaScript on button click.
20. Add and remove classes using `classList` based on user actions (e.g., toggle theme).
21. Attach an event listener using `addEventListener()` to handle button clicks.
22. Demonstrate event handling using mouse events (`mouseover`, `mouseout`), keyboard events (`keydown`, `keyup`), and form events (`submit`, `focus`, `blur`).
23. Create a login or registration form and validate:
 - Required fields
 - Email format
 - Password length
24. Show real-time error messages using DOM for form validation.

Information Security								
Course Code: MDC207-2C				Semester:3		Programme: B.Sc.in Cyber Security, Networking & Cloud Computing (Hons.)		
Teaching Scheme				Evaluation Scheme				
Lecture	Tutorial	Practical	Contact Hours	CIE Theory	CIE Practical	ESE Theory	ESE Practical	Total
3	0	1	5	18/50	-	18/50	-	36/100

Course Objectives:

- Basic understanding of CIA principles and basic principles of the symmetric cryptography and techniques with their strengths and weaknesses from perspective of cryptanalysis
- Implement and analyze various substitution techniques
- Achieving confidentiality using Encryption and understanding Block Ciphers and Data Encryption Standards
- To assess and evaluate the Public key cryptography with various Algorithms

Course Learning Outcomes (CLOs):

1. Understand Information Security Principles. Explain the fundamental principles of information security, including confidentiality, integrity, and availability (CIA triad).
2. Identify Security Threats and Vulnerabilities. Recognize various types of security threats and vulnerabilities that can affect information systems and data.
3. Implement Security Controls. Apply appropriate security controls and measures to protect information assets, including access controls, encryption, and firewalls.
4. Conduct Risk Assessments. Perform risk assessments to identify, evaluate, and prioritize risks to information security, and propose mitigation strategies.

Course Content:

Module 1: Introduction to Information Security, Confidentiality Using Symmetric Encryption:

What Is Information Security, Overview of information, Security, Security mindset, Computer Security Concepts (CIA), Threats, Attacks, and Assets, Security Services, Mechanisms and Attacks, Placement of Encryption Function, Traffic Confidentiality, Key, Distribution, Random Number Generation

Module 2: Symmetric Key Cryptography, Public-Key Cryptography And RSA:

Symmetric Cipher Model, Cryptography, Cryptanalysis, Principles of Public-key Cryptosystems, RSA, Key Management in public-key cryptosystem, Diffie-Hellman Key Exchange, Man-in-Middle attack

Module 3: Substitution Techniques and Steganography:

Caesar Cipher, Monoalphabetic Ciphers, Playfair Cipher, One, Time Pad, Transposition Techniques, Steganography, Digital Signatures, Authentication Protocols, Digital Signature Standard

Module 4: Block Ciphers and The Data Encryption Standard:

Simplified DES, Block Cipher Principles, The Data Encryption, Standard, The Strength of DES, Block Cipher Modes of Operation

Textbooks:

1. Cryptography and Network Security by William Stallings (Pearson Education)
2. Cryptography and Network Security Principal and Practices by Atul Kahate(Tata-McGraw-Hill)
3. Cryptography and Network Security by B A Forouzen (Tata-McGraw-Hill)

Reference Books:

1. Information Systems Security, Godbole, Wiley-India
2. Information Security Principles and Practice, Deven Shah, Wiley-India

List of Open-Source Software/learning website:

1. <https://nptel.ac.in/>
2. <http://www.coursera.org/>

Micro Controller & Processor Programming								
Course Code: AEC207-2C				Semester:3		Programme: B. Sc.in Cyber Security, Networking & Cloud Computing (Hons.)		
Teaching Scheme				Evaluation Scheme				
Lecture	Tutorial	Practical	Contact Hours	CIE Theory	CIE Practical	ESE Theory	ESE Practical	Total
2	1	1	5	18/50	-	18/50	-	36/100

Course Objectives:

- **Introduction to Microcontrollers.** Define what microcontrollers are and their applications. Understand the architecture and components of a micro controller.
- **Programming Fundamentals.** Learn basic programming concepts relevant to microcontrollers programming. Familiarize with programming languages commonly used (e.g., C, Assembly).
- **Development Environment Setup.** Set up the necessary software and hardware tools for microcontrollers programming. Understand the use of Integrated Development Environments (IDEs).
- **Input/Output Operations.** Learn how to interface with various input and output devices (sensors, LEDs, motors). Understand digital and analog I/O operations.

Course Learning Outcomes (CLOs):

1. Identify Microcontroller Components. Recognize and describe the key components of microcontrollers and their functions.
2. Write Basic Programs. Develop simple programs using a programming language (e.g., C) for microcontroller applications. Set Up Development Environment
3. Interface with Hardware. Demonstrate the ability to connect and control various input and output devices using microcontrollers.
4. Implement Control Structures. Utilize control structures (conditional statements, loops) effectively in microcontroller programming

Course Content:

Module: I - Introduction to Linux operating system and Commands & Raspberry pi Board and Interfaces:

Introduction to Linux, OS, File system, Linux Basic, Commands, Scripting in Linux, Hardware of Raspberry pi, HDMI, Micro SD card, 3.5mm audio jack USB port, RJ45 connector, GPIO pins.

Module: II - Installing and Configuring Raspbian OS & Python Programming:

Downloading Raspbian OS, Installing OS on SD card, Boot Raspberry pi, Configuring Raspberry pi, Installation, Environment, Variable, Operator, Decision, Making, Loops, Function

Module: III - Basic digital I/O practical with Motor Control:

GPIO Introduction and Mapping, LED, Switch, IR/PIR sensor, LCD module, DC motor, Keypad, DHT11 temperature sensor, Ultrasonic Sensor (HC SR-04), PWM generation, Servo motor, Stepper motor, Relay, Serial interfacing, Motor Control and PWM, Analog Input

and DC Motor Interfacing, Introduction to Analog to Digital Converter (ADC) and Digital Voltmeter, Introduction to DC Motor, DC Motor with Motor Driver IC L293D Interfacing to Arduino, Servo Motor Interfacing, Servo Motor and Arduino, Servo Motor Interfacing and Programming, Pulse Width Modulation (PWM), Introduction to Pulse Width Modulation (PWM) and Fading Using PWM.

Module: IV - Arduino Based LEDs and Project:

Interfacing Basics, LED Interfacing, Interfacing of Single & Multiple LEDs with Arduino, Interfacing 7-Segment LED Display with Arduino, Buzzer Interfacing, Buzzer Interfacing with Arduino, Tone Generation with Arduino, Switch Interfacing, Introduction to Switches & Interfacing Switches with LEDs to Arduino, Multiple Switches and LEDs Interfacing,

Practical List:

1. LED Interfacing with ARDUINO Microcontrollers. ATMEGA328/32/16/8
2. 0 to 9 counters using single seven segment on ARDUINO. ATMEGA328/32/16/8
3. 0 to 99 UP / Down counter using two seven segments on ARDUINO. ATMEGA328/32/16/8
4. "HELLO WORLD" display on 16*2 LCD on ARDUINO. ATMEGA328/32/16/8
5. Moving Message display on 16*2 LCD on ARDUINO. ATMEGA328/32/16/8
6. Push button switch interfacing Pull UP / Pull Down on ARDUINO ATMEGA328/32/16/8
7. 4*4 Matrix keyboard with LCD on ARDUINO. ATMEGA328/32/16/8
8. Internal ADC with POT. With LCD display on ARDUINO ATMEGA328/32/16/8
9. Internal ADC with LM35 Temperature sensor. ATMEGA328/32/16/8
10. Simple DC Motor & Servo motor Forward & Reverse on ARDUINO ATMEGA328/32/16/8
11. External interrupt, LED Blinking on ARDUINO ATMEGA328/32/16/8
12. Serial data transmission on ARDUINO ATMEGA328/32/16/8
13. Serial data receiving using internal interrupt on ARDUINO ATMEGA328/32/16/8
14. Serial data display on LCD. ATMEGA328/32/16/8
15. Internal EEPROM Read & Write on ARDUINO ATMEGA328/32/16/8
16. Introduction to raspberry pi

Textbooks:

1. Embedded Systems Fundamentals with ARM Cortex-M based Microcontrollers By Alexander G Dean
2. Arduino for Musicians By Brent Edstrom
2. Microcontrollers - Your Custom-Tailored Book By TailoredRead

Reference Books:

1. Beginning C for Microcontrollers By Dr. Jack Purdum

Course: Communication Skills III								
Course Code: SEC207-2C				Semester:3		Programme: B.Sc.in Cyber Security, Networking & Cloud Computing (Hons.)		
Teaching Scheme				Evaluation Scheme				
Lecture	Tutorial	Practical	Contact Hours	CIE Theory	CIE Practical	ESE Theory	ESE Practical	Total
2	0	0	2	18/50	-	18/50	-	36/100

Course Objectives

- To emphasize the essential aspects of effective written communication necessary for Professional success.
- To enable the students to adopt strategies for effective reading and writing Skills.
- The course would enhance student's vocabulary, language and fluency. It would also teach the students to deliver professional presentations.

Course Learning Outcomes (CLOs)

At the end of the course, the students will be able to:

1. Develop and Expand Writing Skills through Controlled and Guided Activities.
2. Apply Verbal and Non-Verbal Communication Techniques in
3. Write well-organized and effective business circulars, notices, agenda and minutes.
4. The students will be able to write correctly and properly with special reference to Letter Writing.

Course Content:

Module: I Introduction

Introduction to Communication, Process of Communication and Feedback.

Module: II Types of Communication:

Verbal Communication Oral Communication - (Face to face communication, Telephone, Presentation, Public Speech, Interview, Meeting), Written Communication - (Letters, Memo, Notice, Circular, Report, Minutes), Non-Verbal Communication, Importance of Non-Verbal Communication, Types of Non-Verbal Communication: -Body Language (Facial expression, Eye contact, Posture, Gesture, Energy, Clothing, Appearance, Touch and Behavior associated with it, Para Language (Vocal Characteristic, Silence)

Module: III Business Writing:

Informal Business, Communication letters (Condolence, greeting, email). Notice Writing and Circulars. Agenda and Minutes Writing.

Module: IV Report Writing:

Proposal Writing, Purpose of proposal writing, Types of proposals, Structure of proposals, developing a proposal, Business Report Writing- Features of a business report, Types of

business reports, Preparing for business reports, Styles of reports.

Textbooks:

1. Business Communication by Meenakshi Raman and Prakash Singh, Oxford University Press
2. Communication Skills for Engineers and Scientists by Sangeeta Sharma and Binod Mishra, PHI Learning
3. Effective Technical Communication by M. Ashraf Rizvi, McGraw Hill

Reference Books:

1. Kaul, A. Business Communication. New Delhi: Prentice Hall of India Private Limited.
2. Lesikar, R. V. and Flately, M. E. Basic Business Communication: Skills for Empowering the Internet Generation. New Delhi: Tata McGraw Hill.
3. Mehra, P. Business Communication for Managers. Pearson.

Course: Indian Knowledge System – II								
Course Code: IKS202-2C				Semester:3		Programme: B.Sc. in Cyber Security, Networking & Cloud Computing (Hons.)		
Teaching Scheme				Evaluation Scheme				
Lecture	Tutorial	Practical	Contact Hours	CIE Theory	CIE Practical	ESE Theory	ESE Practical	Total
2	0	0	2	18/50	-	18/50	-	36/100

Course Objectives (COs)

- To develop an understanding of the philosophical foundations and scientific traditions of the Indian Knowledge System and their relevance in the modern world.
- To explore the contributions of ancient India in science, technology, education, and governance.
- To appreciate India's rich cultural, artistic, and literary heritage through its classical texts and traditions.
- To analyze the modern applications and interdisciplinary relevance of traditional Indian knowledge in areas such as health, ecology, and sustainable living.

Course Learning Outcomes (CLOs)

After successful completion of this course, students will be able to:

1. Explain the key philosophical ideas and epistemological frameworks that form the foundation of Indian Knowledge traditions.
2. Describe the major scientific and technological advancements made in ancient India and their impact on modern science.
3. Interpret the cultural and literary contributions of Indian civilization through classical texts, arts, and educational models.
4. Relate traditional Indian knowledge to contemporary disciplines such as health, environment, ethics, and governance for sustainable development.

Course Content

Module I: Foundations of Indian Knowledge System

Introduction to Indian Knowledge System (IKS) – Definition, Scope, and Relevance; Vedic Vision and Worldview; Sources of Knowledge – Shruti, Smriti, and Anubhav; Six Systems of Indian Philosophy (Darshanas); Indian Epistemology and Logic (Pramana); Concepts of Dharma, Karma, and Moksha; Indian concept of consciousness and holistic living.

Module II: Education, Science and Technology in Ancient India

Ancient Indian education system – Gurukula tradition, Takshashila, Nalanda, and Vikramshila Universities; Teaching–Learning methodologies; Indian contributions in Mathematics (Zero, Decimal System, Geometry), Astronomy (Aryabhata, Varāhamihira), Metallurgy, Chemistry, Architecture, Medicine (Ayurveda), and Yoga; Integration of theory and practice in Indian sciences.

Module III: Art, Culture, and Literature in Indian Civilization

Indian art and aesthetics: Classical dance, music, painting, and sculpture traditions; Literature through the ages – Vedas, Upanishads, Ramayana, Mahabharata, Kalidasa's works, Sangam

literature; Languages and scripts of ancient India; Cultural unity in diversity; Ethical and value dimensions in Indian art and literature.

Module IV: Contemporary Relevance of Indian Knowledge System

Application of IKS in modern society – Holistic health (Yoga, Ayurveda, and meditation), Environmental sustainability (traditional ecological knowledge), Agriculture (Vrikshayurveda), Water management, and Architecture (Vastu Shastra); Governance and ethics based on Arthashastra and Dharma Shastra; Role of IKS in Innovation, Entrepreneurship, and Nation Building; Reinterpreting IKS for global peace and sustainable development.

Text Books

1. “Introduction to Indian Knowledge System: Concepts and Applications” by B. Mahadevan, B. N. Jagdale, and K. Balasubramanian – (AICTE, PHI Learning, 2022)
2. “Essence of Indian Knowledge Tradition” by Dr. S. Kannan and Dr. Sanjeev Kumar – (Mittal Publications, 2021)

Reference Books

1. “The Discovery of India” by Jawaharlal Nehru – Oxford University Press
2. “The Wonder That Was India” by A. L. Basham – Picador India
3. “Indian Philosophy” (Vol. I & II) by Dr. S. Radhakrishnan – Oxford University Press
4. “Indian Knowledge Systems: A Perspective” by Kapil Kapoor and Avadesh Kumar Singh – D.K. Printworld
5. “Science and Technology in Ancient India” by Debiprasad Chattopadhyaya – People’s Publishing House

B.Sc. Animation, VFX & Gaming

Batch 2023-2027

FIFTH SEMESTER

Effective from the Academic Year 2023-24

1.1 B.Sc Animation, Vfx & Gaming Programme Objectives (PO)

- PO1** Apply foundational knowledge of art, design principles, and storytelling to create compelling animation and VFX.
- PO2** Analyze and solve creative and technical problems in animation, VFX, and game development workflows.
- PO3** Design and develop 2D/3D digital content for animation, visual effects, and interactive gaming environments.
- PO4** Use industry-standard software and hardware tools proficiently for production pipelines (e.g., Maya, Nuke, Unreal).
- PO5** Integrate artistic creativity with technical skills to deliver engaging multimedia experiences.
- PO6** Collaborate effectively in multidisciplinary teams adhering to professional ethics and standards.
- PO7** Apply principles of animation, physics, and human anatomy to enhance realism and stylization.
- PO8** Conduct research and stay updated with emerging technologies in animation, VFX, and gaming domains.
- PO9** Manage projects efficiently, including planning, time management, and quality assurance in digital production.
- PO10** Communicate design ideas, workflows, and feedback clearly through visual presentations and documentation.
- PO11** Adapt and innovate using new tools and techniques to meet evolving industry demands.
- PO12** Demonstrate lifelong learning mindset to continuously improve skills and knowledge in the digital media industry.

Syllabus Scheme

- **B.Sc. Animation, Vfx & Gaming (4 years)**
- **Batch 2023-2027**
- **Fifth SEMESTER**

Sr. No	Course Code	Course Name	Circular Components	Teaching Scheme					Exam Scheme		
				L	T	P	Credits	Contact Hours	CCE	SEE	Total
1		Rotoscopy & Paint	Major	0	3	2	4	4	50	50	100
2		Anatomy & Digital Sculpting-I	Major	0	3	2	4	4	50	50	100
3		Tracking & Matchmove	Major	0	3	2	4	4	50	50	100
4		Digital Hair & Groom	Minor	0	3	2	4	4	50	50	100
5		Cloth & Fabric Study	SEC	0	0	4	2	2	25	25	50
6		Vfx Production Elements	Minor	0	3	2	4	4	50	50	100
	TOTAL						22		275	275	550

1.1 B.Sc Animation, Vfx & Gaming Semester-IV: Detail Syllabus

Course: Rotoscopy & Paint								
Course Code:		Credit: 4		Semester: V		Program:B.Sc Animation, Vfx& Gaming		
TeachingScheme				EvaluationScheme				
Lecture	Tutorial	Practical	Contact Hours	CCE Theory	CCE Practical	SEE Theory	SEE Practical	Total
0	3	2	4	18/50	-	18/50	-	36/100

Course Objectives:

1. Recognize the ways to create Roto & Paint subject oriented students

2. Understand the process of Rotoscopy, Paint, Cleanup techniques.
3. Demonstrate through live sessions and different artist works and workshops.
4. Different assignments and discussions with artist will let students explore and analyze more to the utility of creating.
5. Proper evaluation after hand on exercises will guide them to develop Problem-Solving and Artistic Decision-Making Skills.
6. Develop a Professional Portfolio as per standard requirement.

Course Outcomes:

1. **Demonstrate proficiency in using industry-standard software tools**
Students will be able to efficiently use software like **Nuke, Silhouette, or After Effects** for rotoscoping and digital paint tasks within a professional VFX pipeline.
2. **Create high-quality rotoscoping mattes and masks**
Students will produce clean and accurate mattes for complex elements such as hair, motion blur, and semi-transparent surfaces, meeting production-quality standards.
3. **Apply digital paint techniques for shot cleanup**
Students will perform wire/rig removal, background clean-up, and artifact corrections using frame-by-frame painting and procedural techniques.
4. **Analyze and optimize workflows for roto and paint tasks**
Students will demonstrate the ability to evaluate footage and select the most effective strategies and tools to complete shots efficiently and accurately.
5. **Compile a professional-quality showreel demonstrating rotoscoping and paint skills**
Students will present a portfolio of completed projects that reflect their technical and creative skills, suitable for job applications in the VFX industry.

Unit		Contact Hrs.
Unit 1	Fundamentals of Rotoscopy and Paint <ul style="list-style-type: none"> • Introduction to VFX Pipeline and Role of Roto/Paint • Types of Rotoscopy: Garbage Matte, Holdout Matte, Articulated Roto • Understanding Alpha, RGB, and Matte Outputs • Frame-by-Frame Rotoscoping • Roto Shape Creation: Beziers, Splines, and Points • Edge Feathering, Motion Blur, and Softness • Introduction to Rotoscopy Software: Nuke, Silhouette, After Effects • Paint Basics: Clone, Reveal, Heal Tools • Frame-by-Frame vs. Procedural Paint • Color Matching and Grain Management • Working with Multi-Layered Roto 	25
Unit 2	Intermediate Roto and Paint with Tracking & Clean-Up <ul style="list-style-type: none"> • Motion Tracking for Roto (Point Tracking, Planar Tracking) • Mocha Integration for Roto and Tracking • Stabilization Techniques for Roto Shots • Edge Treatments for Hair, Glass, and Motion Blur • Multi-Pass Rotoscoping for Complex Characters • Wire Removal and Rig Cleanup Techniques 	25

	<ul style="list-style-type: none"> • Removing Markers and Set Extensions • BG Reconstruction and Patch Creation • Matching Grains and Light Wraps Post-Paint • Use of Roto in Compositing Pipelines (Pre-mult, Un-premult) 	
Unit 3	Advanced Rotoscopy and Paint Techniques in Production <ul style="list-style-type: none"> • Advanced Paint Techniques: Time-Based Paint, Warping, and Morphing • Advanced Rotoscoping for Semi-Transparent and Hair Elements • Using Expression-based Animation and Scripting for Roto Optimization • Paint and Roto in Stereo (3D) Projects • Shot Management: Versioning, Naming Conventions, and Delivery Formats • QC (Quality Check) for Roto and Paint: What to Look For • Optimization Techniques for Roto Node Trees • Integration into Nuke Compositing Workflow (Pre-comps, Renders) • Team Collaboration: Shot Distribution, File Sharing, Review Tools (e.g., FTrack, ShotGrid) • Building a Professional Roto-Paint Showreel • Studio Simulation Project / Capstone Task 	25

• **List of Assignments (If Applicable):**

Sr. No.	Name of Assignment
1	Basic Shape Rotoscopy (Unit 1)
2	Multi-Layer Roto with Motion Blur (Unit 1 & 2)
3	Tracker-Assisted Rotoscopy (Unit 2)
4	Wire Removal and Clean Plate Paint (Unit 2 & 3)
5	Final Studio Simulated Project (Unit 3)
6	
7	
8	
9	
10	

• **List of Experiments (If Applicable):**

Sr. No.	Name of Experiment
1	
2	
3	
4	
5	
6	

7	
8	
9	
10	

• **CO & PO Mapping Table:**

Course Outcomes (COs)	PO1	PO2	PO3	PO4	PO5	PO6	PO10	PO12
1. Demonstrate proficiency in using industry-standard software tools	3	2	2	–	3	–	–	–
2. Create high-quality rotoscoping mattes and masks	3	3	3	–	2	–	–	–
3. Apply digital paint techniques for shot cleanup	3	3	3	3	3	–	–	–
4. Analyze and optimize workflows for efficient and accurate roto/paint tasks	2	3	2	2	3	–	–	–
5. Compile and present a professional-quality showreel of roto and paint work	2	2	3	–	2	3	3	2

ReferenceBook:

The Art and Science of Digital Compositing

Author: Ron Brinkmann

Publisher: Morgan Kaufmann

ISBN: 9780123706386

Digital Compositing for Film and Video

Author: Steve Wright

Publisher: Routledge

ISBN: 9781138777935.

Rotoscoping: Techniques and Tools for the Aspiring Artist

Author: Benjamin Bratt

Publisher: Focal Press

ISBN: 9781138775795

The VES Handbook of Visual Effects

Editor: Jeffrey A. Okun & Susan Zwerman

Publisher: Focal Press

ISBN: 9780240812427

Nuke 101: Professional Compositing and Visual Effects

Author: Ron Ganbar

Publisher: Peachpit Press

ISBN: 9780321984122

Compositing Visual Effects: Essentials for the Aspiring Artist

Author: Steve Wright

Publisher: Focal Press

ISBN: 9780240813097

Course: Anatomy & Digital Sculpting-I								
Course Code:		Credit: 4		Semester: V		Program:B.Sc Animation, Vfx& Gaming		
TeachingScheme				EvaluationScheme				
Lecture	Tutorial	Practical	Contact Hours	CCE Theory	CCE Practical	SEE Theory	SEE Practical	Total
0	3	2	4	18/50	-	18/50	-	36/100

Course Objectives:

1. Foster **multidisciplinary learning** by integrating anatomy knowledge with digital sculpting skills.
2. Develop **critical observation** and **creative thinking** through anatomical studies and 3D modeling.
3. Promote **hands-on experiential learning** using industry-standard sculpting software (e.g., ZBrush).
4. Encourage **application-based understanding** of human anatomy for realistic and stylized character creation.
5. Enhance **digital literacy** and technical proficiency in modern sculpting tools.
6. Cultivate **ethical and culturally sensitive** artistic representations of human forms.
7. Support **lifelong learning** by building strong foundational skills for future specialization.
8. Encourage **collaborative and peer-learning** through studio practice and group critiques.
9. Promote **self-directed learning** and reflective practices through project-based assignments.

Course Outcomes:

1. Identify and describe key anatomical structures relevant to character design and animation.
2. Demonstrate proficiency in basic digital sculpting tools and software interfaces.
3. Create accurate 3D anatomical models of human body parts such as skull, torso, and limbs.
4. Apply anatomical knowledge to develop both realistic and stylized digital sculptures.
5. Critically analyze and refine their sculptural work through iterative processes and peer feedback.

Unit		Contact Hrs.
Unit 1	Fundamentals of Human Anatomy for Artist	25
	• Overview of human skeletal system: major bones and proportions	
	• Muscular system basics: major muscle groups and their functions	
	• Anatomical landmarks and surface anatomy for character modeling	
	• Proportions and gesture drawing for dynamic poses	
	• Study of human body types and variations	
	• Introduction to anatomy references and resource gathering	

Unit 2	Digital Sculpting using Anatomy <ul style="list-style-type: none"> • Overview of digital sculpting software (ZBrush, Mudbox) and UI navigation • Basic sculpting tools: brushes, alphas, masking, symmetry • Blocking out basic forms: creating skull, torso, limbs using primitives • Understanding volume, flow, and anatomy forms in 3D • Sculpting simple muscles and anatomical details • Introduction to subdivision levels and mesh topology basics • Exporting and saving digital sculptures for workflows 	25
Unit 3	Advanced Anatomical Sculpting and Refinement <ul style="list-style-type: none"> • Sculpting complex muscle groups and anatomical structures (face, hands, feet) • Detailing skin textures and surface features (wrinkles, pores) • Using reference overlays and anatomical planes for accuracy • Sculpting stylized anatomy: exaggeration without losing believability • Introduction to sculpting for animation: joint deformation considerations • Advanced mesh management: Dynamesh, ZRemesher, and retopology basics • Preparing models for rigging and animation pipelines • Critique and iterative refinement techniques for production quality 	25

• **List of Assignments (If Applicable):**

Sr. No.	Name of Assignment
1	Detailed Skeletal Study Sculpt Unit: 1 & 2
2	Muscular Anatomy Sculpt with Surface Detail Unit: 2 & 3
3	Stylized Human Character Sculpt Unit: 3
4	Anatomical Model Preparation for Animation Unit: 3
5	Critique and Iterative Refinement Project Unit: 3
6	

• **List of Experiments (If Applicable):**

Sr. No.	Name of Experiment
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	

• **CO & PO Mapping Table:**

Course Outcomes (COs)	PO1	PO2	PO3	PO4	PO5	PO6	PO10	PO12
1. Identify and describe key anatomical structures relevant to character design	3	2	2	–	–	–	–	–
2. Demonstrate proficiency in basic digital sculpting tools and software interfaces	3	3	2	2	3	–	–	–
3. Create accurate 3D anatomical models of human body parts	3	3	3	2	3	–	–	–
4. Apply anatomical knowledge to develop realistic and stylized digital sculptures	2	3	3	3	3	–	–	–
5. Critically analyze and refine sculptural work through iterative processes	2	3	3	3	3	2	3	2

ReferenceBook:

1. Anatomy for Artists: A New Approach to Discovering, Learning and Remembering the Body

Author: Dr. S. K. Suman **Publisher:** CBS Publishers & Distributors (India)

2. Digital Sculpting with Mudbox: Essential Tools and Techniques for Artists

Authors: Mike de la Flor, Bridgette Mongeon **Publisher:** Wiley India

3. Character Design and Animation Fundamentals

Author: Rachana Ranade (Indian author, educator)

Publisher: CreateSpace Independent Publishing Platform

Course: Tracking & Matchmove			
Course Code:	Credit: 4	Semester:V	Program:B.Sc Animation, Vfx&

				Gaming				
Teaching Scheme				Evaluation Scheme				
Lecture	Tutorial	Practical	Contact Hours	CCE Theory	CCE Practical	SEE Theory	SEE Practical	Total
0	3	2	4	18/50	-	18/50	-	36/100

Course Objectives:

1. Understand the fundamental principles and workflows of 2D and 3D tracking in visual effects production.
2. Gain proficiency in industry-standard tracking and matchmove software tools (e.g., PFTrack, SynthEyes, Mocha).
3. Analyze footage to identify tracking points and solve common tracking challenges like occlusion and motion blur.
4. Apply 3D camera tracking techniques to integrate CG elements seamlessly into live-action footage.
5. Develop skills to create accurate matchmove data for compositing and animation pipelines in VFX projects.

Course Outcomes:

1. Demonstrate a clear understanding of 2D and 3D tracking concepts and workflows in VFX.
2. Operate key tracking and matchmove software tools to solve real-world tracking challenges.
3. Extract and refine tracking data from footage for precise camera solves and object tracking.
4. Integrate 3D elements accurately into live-action scenes using matchmove techniques.
5. Collaborate effectively within a VFX pipeline, delivering tracking data suitable for compositing and animation.

Unit		Contact Hrs.
Unit 1	Fundamentals of 2D and 3D Tracking <ul style="list-style-type: none"> • Introduction to tracking and matchmove in VFX workflows • Principles of 2D tracking: point tracking, planar tracking, and motion tracking • Introduction to 3D tracking: camera solving and tracking markers • Software overview: PFTrack, SynthEyes, Mocha basics • Challenges in tracking: motion blur, occlusion, parallax • Manual vs automatic tracking techniques 	25

	<ul style="list-style-type: none"> Practice: Basic 2D tracking and camera solve exercises 	
Unit 2	Advanced 3D Matchmove Techniques <ul style="list-style-type: none"> Advanced camera solving and refining 3D tracking data Object tracking and multi-object matchmove Handling complex shots: moving cameras, dynamic objects, and variable lens distortion Lens distortion correction and camera calibration Integration of matchmove data with 3D software (Maya or Blender) Use of tracking data for accurate CG object placement and animation Case studies of advanced tracking in professional VFX shots 	25
Unit 3	Workflow Integration and Problem Solving <ul style="list-style-type: none"> Exporting tracking data to compositing and 3D animation pipelines Troubleshooting common tracking errors and refining data Combining multiple tracking techniques for complex shots Use of motion capture data with matchmove for hybrid workflows Tracking for VR/AR applications and real-time pipelines Project: End-to-end matchmove workflow on a complex shot Critique and iterative refinement based on peer and instructor feedback 	25

• **List of Assignments (If Applicable):**

Sr. No.	Name of Assignment
1	Basic 2D Tracking and Stabilization
2	3D Camera Tracking and Solve

3	Object Tracking with Occlusion Handling
4	Lens Distortion Correction and Camera Calibration
5	End-to-End Matchmove Workflow Project

• **List of Experiments (If Applicable):**

Sr. No.	Name of Experiment
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	

• **CO & PO Mapping Table:**

Course Outcomes (COs)	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO9	PO10	PO11
1. Understand 2D and 3D tracking concepts and workflows in VFX.	3	2	2	2	2	–	2	–	–	–
2. Operate tracking and matchmove software tools to solve tracking challenges.	3	3	3	2	3	–	2	–	–	–
3. Extract and refine tracking data for accurate camera solves and object tracking.	3	3	3	2	3	–	3	2	2	2
4. Integrate 3D elements into live-action scenes using matchmove techniques.	3	3	3	2	3	–	3	3	2	2
5. Collaborate and deliver tracking data suitable for compositing and animation.	2	3	3	3	2	3	2	2	3	3

ReferenceBook:

1. Digital Compositing for Film and Video

Author: Steve Wright **Publisher:** Focal Press

2. Matchmoving: The Invisible Art of Camera Tracking

Author: Tim Dobbett **Publisher:** Wiley.

3. The VES Handbook of Visual Effects

Editor: Jeffrey A. Okun & Susan Zwerman **Publisher:** Focal Press

4. Visual Effects and Compositing

Author: Jon Gress **Publisher:** New Riders

5. Fundamentals of VFX and Compositing

Author: Anil Zankar (Indian Author) **Publisher:** Oxford University Press

Course: Digital Hair & Groom								
Course Code:		Credit: 4		Semester:V		Program:B.Sc Animation, Vfx& Gaming		
TeachingScheme				EvaluationScheme				
Lecture	Tutorial	Practical	Contact Hours	CCE Theor y	CCE Practical	SEE Theory	SEE Practical	Total
0	3	2	4	18/50	-	18/50	-	36/100

Course Objectives:

1. To understand the fundamentals of hair and fur structure, behavior, and grooming principles in 3D environments.
2. To introduce students to industry-standard grooming tools such as **XGen**, **Yeti**, **Ornatrix**, or **Houdini Hair**.
3. To develop skills in creating realistic and stylized hair, fur, and feathers for characters and creatures.
4. To explore hair dynamics, simulation, and rendering workflows suitable for film, gaming, and animation.
5. To enhance artistic and technical decision-making in grooming through reference studies and project-based practice

Course Outcomes:

1. **Understand** the anatomy and physical characteristics of hair, fur, and feathers for digital grooming.
2. **Operate** grooming tools (e.g., XGen, Yeti, Houdini Hair) to create realistic and stylized grooms.
3. **Design and develop** groom setups for characters, animals, and creatures suitable for animation and VFX.
4. **Simulate and render** dynamic hair and fur systems with attention to realism and performance.
5. **Analyze and refine** grooming outputs based on feedback, artistic vision, and production requirements.

Unit		Contact Hrs.
Unit 1	Fundamentals of Digital Grooming	20
	• Introduction to hair, fur, and feather anatomy and behavior	
	• Visual analysis: reference gathering for human, animal, and fantasy characters	
	• Principles of grooming: clumping, breakup, layering, and silhouette shaping	

	<ul style="list-style-type: none"> • Tools overview: Maya XGen (Core & Interactive), Yeti, Ornatix, Houdini Groom • Creating basic grooms for: <ul style="list-style-type: none"> • Male characters: short, medium, and stylized hairstyles • Female characters: long hair, braids, ponytails, layered cuts • Accessory hair systems: eyebrows, eyelashes, mustaches, beards • UV and scalp preparation for grooming workflows 	
Unit 2	Advanced Hair, Fur & Animal Grooming <ul style="list-style-type: none"> • Grooming fur for animals: varying length, patching, and stylization • Feather creation for birds, fantasy creatures (procedural and manual methods) • Advanced grooming tools: guide curves, masks, modifiers, texture-based control • Working with clump modifiers, noise patterns, and flyaway hair • Accessory grooming: braids, beards, ornamental hair, horn fur, tail grooming • Style vs. realism: adjusting groom for stylized games vs photoreal VFX • Grooming workflow for hybrid creatures (humanoid-animals) 	20
Unit 3	Simulation, Hair Cards & Game-Ready Workflow <ul style="list-style-type: none"> • Introduction to hair simulation and dynamics (nHair, XGen dynamics, Houdini) • Collision setup, stiffness, drag, and caching systems • Hair card generation for games: <ul style="list-style-type: none"> • Creating optimized hair using cards (polygon planes with hair textures) • Generating alpha maps and normal maps from high-resolution grooms • Baking hair textures for real-time engines (Unreal, Unity) • Placement of cards for male/female game characters • Rendering hair and fur in Arnold, Redshift, and real-time engines • Optimization techniques for grooming in games and VFX • Final project: Groom full character with fur/hair/beard/feathers & simulate/render or export for game engine 	20

- **List of Assignments (If Applicable):**

Sr. No.	Name of Assignment
1	Grooming Male Human Hair (Short Style)
2	Grooming Female Long Hair (Braided or Layered)
3	Eyebrows, Eyelashes, and Beard Grooming
4	Full Animal Fur Groom (Tiger, Lion, or Fox)
5	Feather Grooming for a Bird or Fantasy Creature
6	Hair Dynamics and Simulation
7	Hair Shader and Lighting Test
8	Game Hair Card Creation
9	Stylized Groom for a Game Character
10	Final Project – Full Groomed Character or Creature

- **List of Experiments (If Applicable):**

Sr. No.	Name of Experiment
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	

- **CO & PO Mapping Table:**

Course Outcomes (COs)	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO9	PO10	PO11
1. Understand the anatomy and characteristics of hair, fur, and feathers	3	2	2	–	2	–	3	–	–	–
2. Operate grooming tools like XGen, Yeti, Houdini for various hair/fur types	3	3	3	3	3	–	2	–	–	–
3. Create stylized and realistic grooms for humans, animals, and accessories	3	3	3	3	3	–	3	2	2	2
4. Simulate and render dynamic hair systems for film and animation pipelines	3	3	3	3	3	–	2	2	2	3
5. Generate game-ready hair cards for real-time applications (Unreal/Unity)	3	3	3	3	3	–	2	3	2	3

Reference Book:

Digital Hair and Fur: Using Cg Tools for Hair and Fur Simulation

Author: Gerry Vazquez **Publisher:** New Riders

2. Photorealistic Hair & Fur in CG (*eBook & Guide*) **Author:** Kelly L. Murdock

3. Houdini Hair and Fur (*Online Resource & Booklet*) **Author:** Varomix

4. Game Character Development with Maya **Author:** Antony Ward
Publisher: Focal Press

Course: Cloth & Fabric Study								
Course Code:		Credit: 2		Semester: V		Program:B.Sc Animation, Vfx& Gaming		
TeachingScheme				EvaluationScheme				
Lecture	Tutorial	Practical	Contact Hours	CCE Theory	CCE Practical	SEE Theory	SEE Practical	Total
0	0	4	2	9/25	-	9/25	-	18/50

Course Objectives:

1. To understand the physical properties and behavior of various fabrics and materials used in character and scene design.
2. To introduce students to digital cloth creation techniques and workflows in software like Marvelous Designer, Maya nCloth, and Houdini.
3. To develop practical skills in garment modeling, pattern drafting, and cloth setup for animation and VFX.
4. To explore cloth simulation techniques including collision, wind, gravity, and constraint systems.
5. To prepare students for production-level cloth simulation, optimization, and integration in 3D pipelines for films, games, and virtual production.

Course Outcomes:

1. **Identify** and analyze the physical properties of different types of fabrics and their impact on cloth behavior.
2. **Create** accurate 3D fabric models and garment patterns using industry-standard software like Marvelous Designer and Maya nCloth.
3. **Simulate** realistic cloth movement and interaction, including collision detection and response with characters and environments.

4. **Optimize** cloth simulations for efficient rendering and real-time applications in animation, VFX, and gaming.
5. **Integrate** cloth simulations seamlessly into animation pipelines, ensuring production-quality results suitable for film and game projects.

Unit		Contact Hrs.
Unit 1	Fundamentals of Cloth and Fabric Properties <ul style="list-style-type: none"> • Introduction to textile types: natural vs synthetic fibers • Physical and mechanical properties of fabrics: weight, stretch, stiffness, drape • Understanding weave patterns and fabric construction • Basics of garment anatomy and structure • Overview of cloth simulation concepts: particles, forces, constraints • Introduction to cloth simulation tools: Marvelous Designer, Maya nCloth, Houdini Cloth • Creating simple fabric swatches and testing physical behavior in simulations 	20
Unit 2	Traditional and Western Clothing Study and Simulation Study of traditional clothing styles (e.g., sarees, dhotis, kimonos, togas) — fabric types and draping techniques <ul style="list-style-type: none"> • Simulation challenges with traditional garments: long flowing fabrics, pleats, folds • Western clothing styles: shirts, trousers, dresses, jackets — garment construction and fabric choices • Pattern drafting and 3D garment creation for both traditional and western clothes • Simulating complex garment behaviors: layering, collision, wind effects • Case studies of cloth simulation in film and animation involving traditional and western costumes 	20
Unit 3	Accessories, Advanced Cloth Simulation & Exporting for Games <ul style="list-style-type: none"> • Study and simulation of fabric-based accessories: scarves, gloves, hats, veils • Materials and behavior of leather, silk, lace, fur, and other specialized textiles • Advanced cloth simulation techniques: constraints, sewing patterns, collision, friction, layering • Optimization techniques for efficient cloth simulation in real-time environments 	20

	<ul style="list-style-type: none"> • Integration of cloth simulation with character rigs and animation (skin binding, joint influence) • Exporting cloth simulations to game engines: <ul style="list-style-type: none"> • Baking simulation caches • Creating collision meshes compatible with Unreal Engine and Unity • Converting cloth animation to vertex animation textures (VAT) or blend shapes • Setting up cloth physics assets in Unreal Engine (Chaos Cloth, Nvidia Apex Cloth) • Performance optimization for real-time playback without compromising quality • Final project: Complete cloth simulation with accessories exported and set up in Unreal Engine for real-time preview 	
--	---	--

• **List of Assignments (If Applicable):**

Sr. No.	Name of Assignment
1	Automating Basic Rigging Tasks with Python or MEL"
2	Creating an IK-FK Limb Rig with Seamless Switching"
3	Building a Stretchy Spine Rig with Advanced Controls"
4	Designing a Facial Rig with Blendshapes and GUI Controls"
5	Rigging a Quadruped for Realistic Locomotion"
6	Dynamic Rigging for Tails, Wings, or Tentacles"
7	Integrating Muscle Systems for Realistic Skin Deformation"
8	Exporting Game-Ready Rigs Compatible with Unreal Engine"
9	Custom Spline IK Rigging for Ropes or Flexible Props"
10	Final Project: Advanced Rig for a Complex Character or Creature"

• **List of Experiments (If Applicable):**

Sr. No.	Name of Experiment
1	
2	
3	
4	
5	
6	
7	
8	

9	
10	

• **CO & PO Mapping Table:**

Course Outcomes (COs)	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO9	PO10	PO11
1. Understand physical and mechanical properties of various fabrics	3	2	2	–	2	–	3	–	–	–
2. Create and simulate traditional and western garments	3	3	3	3	3	–	2	–	2	–
3. Simulate fabric-based accessories with realistic behavior	3	3	3	3	3	–	2	–	2	–
4. Optimize cloth simulations for real-time applications	3	3	3	3	3	–	3	2	2	2
5. Export and integrate cloth simulations into game engines (Unreal etc.)	3	3	3	3	3	–	2	3	3	3

Reference Book:

- **Marvelous Designer: Garment Modeling & Simulation**
Author: Gabriel Salas
- **Cloth Simulation for Computer Graphics**
Authors: Helmut Pottmann, et al.
- **Digital Garment Design: 3D Modeling, Simulation, and Rendering**
Author: Anupam Saxena (Indian Author)

Course: Vfx Production Elements								
Course Code:		Credit: 4		Semester:V		Program:B.Sc Animation, Vf x& Gaming		
TeachingScheme				EvaluationScheme				
Lecture	Tutorial	Practical	Contact Hours	CCE Theor y	CCE Practical	SEE Theory	SEE Practical	Total
0	3	2	4	18/50	-	18/50	-	36/50

Course Objectives:

1. To introduce students to the end-to-end process of producing visual effects elements for film, television, and digital media.
2. To develop practical skills in planning, executing, and managing VFX production workflows and pipelines.
3. To enable students to integrate various VFX disciplines such as compositing, 3D tracking, simulation, and matte painting into a unified project.

4. To foster teamwork and communication skills through collaborative project work simulating real-world VFX production environments.
5. To cultivate problem-solving abilities for addressing technical and artistic challenges during VFX production.

Course Outcomes:

1. Apply the principles of visual effects production to plan, design, and execute a complete VFX element or sequence.
2. Demonstrate proficiency in integrating multiple VFX components such as compositing, tracking, modeling, and simulation into a cohesive final output.
3. Manage production pipelines effectively, ensuring technical and artistic quality while adhering to deadlines.
4. Collaborate with multidisciplinary teams to solve creative and technical challenges in VFX projects.
5. Critically analyze and troubleshoot VFX workflows to optimize performance and final render quality for various media platforms.

Unit		Teaching Hrs.
Unit 1	Shot PREPRODUCTION Prepare a preproduction through PPT and Mood board and finalizing the concept, story, storyboard, Character design.	25
Unit 2	Shot PRODUCTION Create final project using 3d Tools and finalizing through passes and final rendering	25
Unit 3	Portfolio and Presentation The student will be given a chance to interact with the Industry Experts and they will give students what actually the Industry and how professionals work. The student will have to submit a project report regarding all the knowledge they have gained through interaction with professionals and finally showcasing the movie through editing and sound of final film.	25

• **List of Assignments (If Applicable):**

Sr. No.	Name of Assignment
1	
2	
3	
4	
5	
6	

7	
8	
9	
10	

- List of Experiments (If Applicable):**

Sr. No.	Name of Experiment
1	Preproduction of Shot
2	Production of Shot
3	Post Production of Shot
4	Titling and final Presentation documentation of the entire process.

- CO & PO Mapping Table:**

Course Outcomes (COs)	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO9	PO10	PO11
1. Apply VFX production principles to plan and execute VFX elements	3	3	3	3	3	2	2	3	3	3
2. Integrate compositing, tracking, simulation, and other VFX components	3	3	3	3	3	2	2	3	3	3
3. Manage VFX production pipeline ensuring quality and deadlines	3	3	3	3	3	3	2	3	2	2
4. Collaborate effectively in multidisciplinary teams	2	3	3	2	2	3	3	2	3	3
5. Analyze and troubleshoot VFX workflows for optimization	3	3	3	3	3	2	3	3	3	3

Reference Book:
VFX Pipeline

Course: Analysis of Design And Algorithm								
Course Code:		Credit:4		Semester: III			Programme: MCA	
Teaching Scheme				Evaluation Scheme				
Lecture	Tutorial	Practical	Contact Hours	CIE Theory	CIE Practical	ESE Theory	ESE Practical	Total
3	0	2	5	25/50	-	25/50	-	50/100

Course Objectives:

- Analyze algorithm complexity using asymptotic notations
- Implement sorting and searching algorithms effectively
- Apply divide-and-conquer and dynamic programming strategies for optimization
- Utilize greedy algorithms for problem-solving in scheduling and spanning trees & Implement graph traversal techniques using BFS, DFS, and topological sorting.

Course Learning Outcomes (CLOs):

By the end of the course, students will be able to:

1. Demonstrate the ability to analyze the time and space complexity of algorithms using appropriate notations.
2. Apply sorting, searching, and optimization techniques to solve computational problems efficiently.
3. Implement algorithmic strategies such as divide and conquer, dynamic programming, and graph traversal in practical scenarios.
4. Apply backtracking, branch and bound, and string matching algorithms to solve optimization and pattern matching problems.

Course Content

Module I: Foundations of Algorithm Analysis (28%)

Basic concepts of algorithms and their analysis, Time and space complexity, Asymptotic notation (Big-O, Omega, Theta), Best, worst, and average case analysis, Sorting and Searching Algorithms: Selection sort, Bubble sort, Insertion sort, Shell sort, Heap sort, Sorting in linear time: Bucket sort, Radix sort, Counting sort

Module II: Divide and Conquer & Dynamic programming (30%)

Divide and conquer techniques: Quick sort, Merge sort, Matrix multiplication, Binary search, Binary search tree, Knapsack problem, All Points Shortest path, Matrix chain multiplication, Longest Common Subsequence, Introduction to Dynamic programming, Principle of Optimality, Problem Solving using Dynamic programming, Calculating the Binomial Coefficient, Making Change Problem, Assembly Line Scheduling, Dynamic programming: Knapsack problem, All Points Shortest path, Matrix chain multiplication, Longest Common Subsequence

Module III: Greedy Algorithms & Graph Theory (25%)

General Characteristics of Greedy Algorithms, Problem Solving using Greedy Algorithms: Activity Selection Problem, Elements of Greedy Strategy, Minimum Spanning Trees (Kruskal's algorithm, Prim's algorithm), Graphs: Shortest paths, The Knapsack Problem, Job Scheduling Problem, Huffman code, Graph Fundamentals and Traversal Techniques, Depth First Search, Breadth First Search, Topological Sort

Module: IV- Backtracking, Branch and Bound, and String Matching (17%)

Introduction to Backtracking and Branch and Bound, Eight Queens Problem, Knapsack Problem, Travelling Salesman Problem, Minimax Principle, String Matching Algorithms, Naive String Matching, Rabin-Karp Algorithm, Knuth-Morris-Pratt Algorithm, String Matching with Finite Automata

Text Books:

1. "Introduction to Algorithms" by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein - Comprehensive coverage of algorithm design and analysis techniques, including sorting, searching, dynamic programming, and graph algorithms.
2. "Algorithm Design" by Jon Kleinberg and Éva Tardos - Focuses on algorithm design techniques such as greedy algorithms, divide and conquer, and dynamic programming.
3. "Data Structures and Algorithm Analysis in C++" by Mark Allen Weiss - Provides in-depth coverage of algorithm analysis and implementation with C++.
4. "Design and Analysis of Algorithms" by S. Sridhar - Detailed explanations of algorithmic techniques and problem-solving approaches with examples.
5. "The Algorithm Design Manual" by Steven S. Skiena - Practical guide to algorithm design, implementation, and optimization techniques.

References Books:

1. "Algorithms Unlocked" by Thomas H. Cormen - A beginner-friendly guide to understanding essential algorithm concepts.
2. "Algorithms" by Robert Sedgewick and Kevin Wayne - Comprehensive coverage of algorithms and data structures with real-world applications.
3. "Data Structures and Algorithms Made Easy" by Narasimha Karumanchi - Step-by-step algorithmic problem-solving approach.
4. "Algorithmics: The Spirit of Computing" by David Harel - Concepts of algorithm design and computational problem-solving.
5. "Computer Algorithms: Introduction to Design and Analysis" by Sara Baase and Allen Van Gelder - Classic reference for algorithm analysis and complexity.

List of Suggestive s:

1. Implementation and Time analysis of sorting algorithms.
2. Bubble sort, Selection sort, Insertion sort, Merge sort and Quicksort
3. Implementation and Time analysis of linear and binary search algorithm.
4. Implementation of max-heap sort algorithm
5. Implementation and Time analysis of factorial using iterative and recursive method
6. Implementation of a knapsack problem using dynamic programming.
7. Implementation of chain matrix multiplication using dynamic programming.
8. Implementation of Graph and Searching (DFS and BFS).
9. Implement LCS problem.

PO-COMPETENCY-CLO MAPPING:

For CLO-PO mapping:

CLO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CLO1	H	H	M	M	L	L	M	L	M	L	M	M
CLO2	H	H	H	M	M	L	M	M	H	M	L	M
CLO3	H	H	H	H	M	M	H	L	M	M	H	M
CLO4	H	H	M	M	L	M	M	L	M	L	M	L

For CLO-PSO mapping:

CLO	PSO1	PSO2	PSO3	PSO4
CLO1	H	M	L	M
CLO2	H	H	M	L
CLO3	H	M	H	M
CLO4	M	H	M	L

Course: AI and Machine learning using Python								
Course Code:		Credit:4		Semester: III			Programme: MCA	
Teaching Scheme				Evaluation Scheme				
Lecture	Tutorial	Practical	Contact Hours	CIE Theory	CIE Practical	ESE Theory	ESE Practical	Total
3	0	2	5	25/50	-	25/50	-	50/100

Course Objectives:

- To introduce students to fundamental concepts of Artificial Intelligence (AI) and Machine Learning (ML).
- To explore core algorithms and techniques used in AI and ML.
- To develop the ability to implement AI/ML solutions using Python libraries like Scikit-learn, NumPy, Pandas, and TensorFlow/Keras.
- To understand and apply various ML models for classification, regression, and clustering.

Course Learning Outcomes (CLOs):

By the end of the course, students will be able to:

1. Understand foundational concepts and tools in Artificial Intelligence and Machine Learning, including the history, types of AI, differences between AI, ML, and Deep Learning, and the use of essential Python libraries (NumPy, Pandas, Matplotlib, Seaborn) along with Jupyter Notebooks for data analysis and visualization.
2. Apply supervised and unsupervised learning algorithms such as Linear and Logistic Regression, k-Nearest Neighbors, Support Vector Machines, K-Means, Hierarchical Clustering, and Principal Component Analysis (PCA) to real-world datasets, and evaluate model performance using metrics like Confusion Matrix, Precision, Recall, F1-score, and ROC-AUC.
3. Design and implement decision tree-based models and ensemble methods such as Random Forest, Gradient Boosting, and Bagging techniques for improved predictive accuracy and understanding of the trade-offs between bias and variance.
4. Develop and deploy basic neural networks using TensorFlow/Keras for tasks like image or text classification, demonstrate awareness of ethical considerations such as bias and fairness in AI systems, and explore emerging trends in AutoML, MLOps, and AI applications in various domains like healthcare, finance, and education.

Course Content

Module I: Introduction to AI and Python for ML (30%)

History and evolution of AI, AI applications and domains, Types of AI: Narrow, General, Super, Difference between AI, ML, and Deep Learning, Python libraries for AI/ML: NumPy, Pandas, Matplotlib, Seaborn Introduction to Jupyter Notebooks and data visualization

Supervised Learning

Definition and types (classification and regression), Linear Regression, Logistic Regression, k-Nearest Neighbors (k-NN), Support Vector Machines (SVM), Model evaluation: Confusion Matrix, Precision, Recall, F1-score, ROC-AUC

Module II: Unsupervised Learning (20%)

Clustering: K-Means, Hierarchical Clustering, Dimensionality Reduction: PCA, Applications of unsupervised learning

Decision Trees and Ensemble Methods

Decision Trees and Entropy, Random Forest, Gradient Boosting Machines (GBM), Bagging vs Boosting

Module III: Introduction to Neural Networks and Deep Learning (30%)

Basics of Neural Networks, Perceptron and Multi-layer Perceptron, Backpropagation

Introduction to TensorFlow/Keras, Simple deep learning model for image or text classification

Module IV: Real-world Applications and Ethical Considerations (20%)

AI in healthcare, finance, education, and e-commerce, AI fairness, bias, and explainability,

Introduction to AutoML and MLOps (basic concepts), Future trends in AI

Text Books:

1. **"Python Machine Learning"** by Sebastian Raschka and Vahid Mirjalili
2. **"Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow"** by Aurélien Géron
3. **"Artificial Intelligence: A Modern Approach"** by Stuart Russell and Peter Norvig

References Books:

1. Online resources: [Scikit-learn documentation](#), TensorFlow tutorials

List of suggestive programs:

1. to demonstrate NumPy arrays and vectorized operations.
2. Use Pandas to load and explore the Titanic dataset.
3. Plot histograms, scatter plots, and heatmaps using Matplotlib and Seaborn.
4. Create a notebook to analyze the Iris dataset: summary stats, visualization, and basic filtering.
5. Predict house prices using the Boston Housing dataset.
6. Predict survival on the Titanic dataset (classification problem)
7. Implement k-NN for handwritten digit classification using the digits dataset.
8. Build an SVM classifier for the Iris dataset and visualize decision boundaries.
9. to compare Logistic Regression and SVM using confusion matrix, precision, recall, F1-score, and ROC curve.
10. Apply K-Means to customer segmentation (e.g., using Mall Customers dataset).
11. Dendrogram plotting and customer segmentation.
12. Reduce dimensions of the Iris or Wine dataset and visualize in 2D.
13. Image compression using K-Means.
14. Implement on the Titanic dataset and visualize the tree.
15. Compare Random Forest performance with Decision Tree on classification task.
16. Use Gradient Boosting for binary classification and compare with other models.
17. Compare accuracy of BaggingClassifier and GradientBoostingClassifier on a dataset.
18. Predict diabetes using the Pima Indians Diabetes dataset.
19. Credit card fraud detection using classification algorithms.
20. Predict student performance using a regression model.
21. Use SHAP or LIME to explain predictions made by a classifier.
22. Demonstrate how models perform differently across gender/age groups using fairness metrics.

23. Use TPOT or Auto-sklearn to auto-generate ML pipelines.
24. Use MLflow for tracking experiments and model versioning.
25. Complete ML pipeline from data collection, preprocessing, modeling, evaluation, and deployment using Flask or Streamlit.

PO-COMPETENCY-CLO MAPPING:

For CLO-PO mapping:

CLOs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CLO1	M	H	M	M	L	H	M	H	M	L	M	M
CLO2	H	M	H	H	H	M	H	L	H	M	H	H
CLO3	M	H	M	H	M	L	M	H	M	H	M	M
CLO4	M	H	H	H	H	M	M	L	H	M	M	H

For CLO-PSO mapping:

CLO	PSO1	PSO2	PSO3	PSO4
CLO1	M	H	M	M
CLO2	H	L	M	H
CLO3	M	L	H	M
CLO4	M	H	H	M

Course: Mobile Application Development using Flutter								
Course Code:	Credit:4	Semester: III				me: MCA		
Teaching Scheme				Evaluation Scheme				
Lecture	Tutorial	Practical	Contact Hours	CIE Theory	CIE Practical	ESE Theory	ESE Practical	Total
3	0	2	5	25/50	-	25/50	-	50/100

Course Objectives:

- To introduce students to cross-platform mobile development using Flutter.
- To understand Dart ming and UI components of Flutter.
- To build responsive and interactive mobile apps.
- To integrate apps with back-end services and data storage.

Course Learning Outcomes (CLOs):

By the end of the course, students will be able to:

1. Understand and set up the Flutter development environment and Dart ming fundamentals;
2. Design and develop interactive user interfaces using core Flutter widgets and layout mechanisms;
3. Implement navigation and state management techniques in multi-screen Flutter applications;
4. Integrate mobile apps with external data sources using APIs and manage local storage.

Course Content

Module I: Introduction to Flutter & Dart (20%)

Overview of mobile development platforms, Flutter framework and architecture, Setting up Flutter environment (IDE, SDK), Introduction to Dart: variables, data types, functions, control structures, Object-Oriented features of Dart.

Module: II Flutter Widgets and UI Components (25%)

Understanding widgets: Stateless vs Stateful, Basic widgets: Text, Image, Icon, Button, AppBar, Scaffold, Input and Form widgets: Text Field, Checkbox, Radio Button, Switch, Layout widgets: Container, Row, Column, Stack, List View, Grid View.

Module: III Navigation and State Management (25%)

Navigation using Navigator and routes, Passing data between screens, State management using set State (), lifting state up, Introduction to Provider for state management.

Module: IV Data Handling and Storage (30%)

Working with JSON data and API calls, Fetching data using http package, Local storage using Shared Preferences, Introduction to local database using SQLite.

Text Books:

1. "Beginning Flutter: A Hands-On Guide to App Development" by Marco L. Napoli.
2. "Flutter for Beginners" by Alessandro Biessek.

References Books:

1. "ming Flutter" by Carmine Zaccagnino.
2. Anil Jain K., Fundamentals of Digital Image Processing, PHI Learning.

List of Suggestive s:

1. Set up Flutter environment and create your first Flutter app.
2. Implement basic UI using Stateless and Stateful widgets.
3. Design login form using form widgets and validation.
4. Create multi-screen app with navigation.
5. Fetch and display JSON data from an API.
6. Store and retrieve data using Shared Preferences.
7. Use SQLite to implement local database functionality.
8. Create a complete mobile app as a mini project.

PO-COMPETENCY-CLO MAPPING:

For CLO-PO mapping:

CLO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CLO1	H	M	M	M	L	L	L	M	M	L	L	M
CLO2	H	H	H	H	H	M	M	L	H	M	M	H
CLO3	M	M	H	H	H	M	M	L	M	M	M	H
CLO4	H	M	H	H	H	M	M	L	H	M	M	H

For CLO-PSO mapping:

CLO	PSO1	PSO2	PSO3	PSO4
CLO1	M	H	M	M
CLO2	H	L	H	H
CLO3	M	L	H	M
CLO4	M	H	H	M

Course: Image Processing (Elective-1)								
Course Code:	Credit:4	Semester: III				me: MCA		
Teaching Scheme				Evaluation Scheme				
Lecture	Tutorial	Practical	Contact Hours	CIE Theory	CIE Practical	ESE Theory	ESE Practical	Total
3	0	2	5	25/50	-	25/50	-	50/100

Course Objectives:

- Provide an introduction to the basic concepts and methodologies for digital image processing.
- To develop a foundation that can be used as a basis for further studies and research.
- Introduce students to the core techniques and algorithms involved in the acquisition and processing of digital images.
- Teach methods for extracting meaningful information from digital images through analysis and interpretation.

Course Learning Outcomes (CLOs):

By the end of the course, students will be able to:

1. Get acquainted with digital image fundamentals and its applications and get acquainted with the image representation and description methods;
2. Learn and perform image pre-processing and enhancement to improve the image for further processing;
3. Reconstruct photometric properties degraded by the imaging process and partition a digital image into multiple segments;
4. Represent and analyse images at different resolutions, process images according to their shapes, and apply compression techniques to reduce the storage space of images.

Course Content

Module I: Image Fundamentals (25%)

Introduction to Digital Image Processing and its applications; Components of an Image Processing System.

Image Representation and Description: Image Representation ; Digital Image Properties; Boundary descriptors; Regional descriptors; Steps in Digital Image Processing; Elements of Visual perception; Image Sensing and Acquisition; Image Sampling and Quantization; Relationship between Pixels; Color Representation.

Data Structures for Image Analysis: Levels of Image Data Representation; Traditional Image Data Structures: Matrices, Chains, Topological Data Structures, Relational Structures; Hierarchical Data Structures: Pyramids, Quadrees, Other Pyramidal Structures.

Module II: Image Pre-Processing (25%)

Pixel Brightness Transformations: Position-Dependent Brightness Correction, Gray-Scale Transformation; Geometric Transformations: Pixel Co-ordinate Transformations, Brightness Interpolation; Local Pre-Processing.

Image Enhancement: Spatial Domain: Gray level transformations; Histogram processing; enhancement using arithmetic and logic operators; Basics of Spatial Filtering; Smoothing and

Sharpening Spatial Filtering.

Frequency Domain: Introduction to Fourier Transform; Filtering in the Frequency Domain; Smoothing and Sharpening frequency domain filters; Homomorphic Filtering.

Module: III Image Restoration and Segmentation (25%)

Noise models; Mean Filters; Order Statistics; Adaptive filters; Noise Reduction by Frequency Domain Filtering; Inverse and Wiener filtering; Constrained Least Squares Filtering.

Segmentation: Point, line, and Edge Detection; Edge Linking and Boundary detection; Thresholding; Region based segmentation; Edge based Segmentation; Segmentation by Morphological Watersheds; Matching.

Color Image Processing: Color Fundamentals, Color Models, Pseudocolor Image Processing

Module III: Wavelets and Multiresolution Processing (25%)

Background: Image Pyramids; Subband coding; Multiresolution expansions.

Morphological Image Processing: Preliminaries, Erosion and Dilation, Opening and Closing, The Hit-or-Miss Transforms, Some Basic Morphological Algorithms.

Compression – Fundamentals ; Image Compression models; Error-Free Compression; Variable Length Coding, LZW coding, Bit-Plane Coding, Lossless Predictive Coding; Lossy Compression: Lossy Predictive Coding, Transform Coding, wavelet Coding; Image Compression Standards.

Text Books:

1. Rafael C. Gonzales, Richard E. Woods, Digital Image Processing, Pearson Education.

References Books:

1. Rafael C. Gonzalez, Richard E. Woods, Steven L. Eddins, Digital Image Processing Using MATLAB, Third Edition ,Tata McGraw Hill .
2. Anil Jain K., Fundamentals of Digital Image Processing, PHI Learning.
3. Willlliam K Pratt, Digital Image Processing, John Willey.
4. Malay K. Pakhira, Digital Image Processing and Pattern Recognition, First Edition, PHI Learning.

List of Suggestive s:

1. To introduce digital image processing and demonstrate reading, displaying, and writing images using MATLAB.
2. To perform image representation conversions including grayscale and binary conversions.
3. To analyze and visualize pixel relationships and neighborhood connectivity.
4. To perform image sampling and quantization and observe their effect on image quality.
5. To study image descriptors and extract boundary and region features.
6. To implement and visualize different color models and transformations.
7. To understand and apply different data structures for image analysis like matrices, quadrees, and pyramids.
8. To apply brightness and contrast transformations including negative, logarithmic, and power-law (gamma) functions.
9. To perform histogram equalization and contrast stretching.
10. To apply basic spatial filters for smoothing and sharpening using convolution kernels.
11. To perform image enhancement using arithmetic and logical operations.
12. To implement geometric transformations such as scaling, rotation, and translation.

13. To implement Fourier transform and perform filtering in the frequency domain.
14. To enhance an image using homomorphic filtering techniques.
15. To simulate image degradation by adding noise and restore using mean and median filters.
16. To implement adaptive filtering and frequency-domain noise reduction.
17. To restore images using inverse filtering and constrained least squares filtering.
18. To detect edges using Sobel, Prewitt, and Canny methods and perform boundary linking.
19. To implement thresholding and region-based segmentation techniques.
20. To perform segmentation using morphological watershed algorithm.
21. To process color images using RGB and pseudocolor techniques.
22. To apply multiresolution analysis using image pyramids and wavelet decomposition.
23. To implement basic morphological operations such as erosion, dilation, opening, and closing.
24. To apply advanced morphological algorithms like boundary extraction, hit-or-miss, and thinning.
25. To understand and apply basic image compression techniques including Huffman and LZW coding.
26. To implement transform coding using DCT and wavelet coding for image compression.
27. To compare lossless and lossy compression techniques using quality metrics such as PSNR and compression ratio.

PO-COMPETENCY-CLO MAPPING:

For CLO-PO mapping:

CLO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CLO1	M	H	M	M	L	H	M	H	M	L	M	M
CLO2	H	M	H	H	H	M	H	L	H	M	H	H
CLO3	M	H	M	H	M	L	M	H	M	H	M	M
CLO4	M	H	H	H	H	M	M	L	H	M	M	H

For CLO-PSO mapping:

CLO	PSO1	PSO2	PSO3	PSO4
CLO1	M	H	M	M
CLO2	H	L	M	H
CLO3	M	L	H	M
CLO4	M	H	H	M

Course: Cryptography & Network and Security(Elective-1)								
Course Code:	Credit:4	Semester: III				Programme: MCA		
Teaching Scheme				Evaluation Scheme				
Lecture	Tutorial	Practical	Contact Hours	CIE Theory	CIE Practical	ESE Theory	ESE Practical	Total
3	0	2	5	25/50	-	25/50	-	50/100

Course Objectives:

- Understand core concepts of computer security, security services, mechanisms, and the OSI Security Architecture.
- Apply and analyze symmetric and asymmetric encryption techniques, including classical and block ciphers like DES and AES.
- Evaluate cryptographic hash functions and message authentication techniques for ensuring data integrity and authenticity.
- Explain IP Security (IPSec), its architecture, and identify different types of system and program-level security threats.

Course Learning Outcomes (CLOs):

By the end of the course, students will be able to:

1. Illustrate different types of security attacks, services, and mechanisms using real-life examples and models like Caesar Cipher and Steganography in Python.
2. Demonstrate encryption and decryption using block cipher operations (e.g., ECB, CBC, CFB, OFB, and CTR modes) and understand key distribution methods.
3. Analyze the structure and function of cryptographic hash algorithms such as SHA and use them to create Message Authentication Codes (MACs).
4. Compare IPSec tunnel and transport modes and identify various network threats including DoS, DDoS, worms, viruses, and scanning attacks.

Course Content

Module I: Introduction (25%)

Computer Security Concepts, The OSI Security Architecture, Security Attacks, Security Services, Security Mechanisms, A Model for Network Security, CLASSICAL ENCRYPTION TECHNIQUES: Symmetric Cipher Model, Substitution Techniques, Caesar cipher, Rainfall, Steganography(Python Script)

Module II: Block Ciphers (25%)

Symmetric Key Distribution, Asymmetric Key Distribution, BLOCK CIPHERS AND THE DATA ENCRYPTION STANDARD: Block Cipher Principles, The Data Encryption Standard (DES), AES(Advanced Encryption Standard), Block Cipher Design Principles. BLOCK CIPHER OPERATION: Multiple Encryption and Triple DES, Electronic Codebook Mode, Cipher Block Chaining Mode, Cipher Feedback Mode, Output Feedback Mode, Counter Mode. STREAM CIPHERS : Stream Ciphers

Module III: Cryptographic Hash Functions (25%)

Applications of Cryptographic Hash Function, Two Simple Hash Functions, Hash Functions Based on Cipher Block Chaining, Secure Hash Algorithm (SHA). MESSAGE AUTHENTICATION

CODES : Message Authentication Requirements, Message Authentication Functions, Message Authentication Codes, Digital Signatures

Module IV: IP Security Architecture and System Security (25%)

What is IP Security (IPSec), Importance of IPSec, Features of IPSec, How Does IPSec Work, IPSec Connection Establishment Process, Difference Between IPSec Tunnel Mode and IPSec Transport Mode, Protocols Used in IPSec, IP Security Architecture, IPSec Encryption, IPSec VPN, Advantages of IPSec, Disadvantages of IPSec, Types of Program Threats(Virus, Trojan Horse, Trap Door, Logic Bomb), Types of System Threats(Worm, Port Scanning, DOS, DDOS)

Text Books:

1. **William Stallings**, *Network Security Essentials: Applications and Standards*, 6th Edition, Pearson Education, ISBN: 9780134527338.
2. **William Stallings**, *Cryptography and Network Security: Principles and Practice*, 8th Edition, Pearson Education, ISBN: 9780136687290

References Books:

1. **Behrouz A. Forouzan**, *Cryptography and Network Security*, 1st Edition, McGraw Hill Education, ISBN: 9781259641230.
2. **Charlie Kaufman, Radia Perlman, and Mike Speciner**, *Network Security: Private Communication in a Public World*, 2nd Edition, Pearson Education, ISBN: 9780130460196

Practical List:

1. Implement Caesar Cipher encryption and decryption using Python.
2. Encrypt and decrypt messages using DES symmetric encryption in Python
3. Implement AES encryption and decryption using Python's cryptography library
4. Implement Monoalphabetic Cipher encryption and decryption using Python.
5. Implement CBC cipher in Python
6. Simulate a DoS (Denial of Service) attack in a controlled lab environment.
7. Implement Diffie-Hellman key exchange algorithm using Python.
8. Implementation of Poly alphabetic cipher
9. Implement IPSec VPN single channel.
10. Implement IPSec VPN dual channel.
11. Implement DM VPN .

PO-COMPETENCY-CLO MAPPING:

For CLO-PO mapping:

CLO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CLO1	H	H	M	L	M	L	L	L	L	M	L	L
CLO2	H	H	H	M	H	L	L	L	M	M	M	L
CLO3	H	H	H	M	H	M	L	L	M	M	L	M
CLO4	H	H	M	M	M	M	M	L	M	M	M	M

For CLO-PSO mapping:

CLO	PSO1	PSO2	PSO3	PSO4
CLO1	H	M	L	M
CLO2	H	H	M	M
CLO3	H	H	M	H
CLO4	H	H	H	H

Course: .NET MVC(Elective -1)								
Course Code:		Credit:4	Semester: III			Programme: MCA		
Teaching Scheme				Evaluation Scheme				
Lecture	Tutorial	Practical	Contact Hours	CIE Theory	CIE Practical	ESE Theory	ESE Practical	Total
3	0	2	5	25/50	-	25/50	-	50/100

Course Objectives:

- Understand the MVC architecture, development environment setup, and routing mechanism in ASP.NET MVC.
- Develop MVC applications using controllers, views, partial views, and state management techniques.
- Integrate database using Entity Framework and implement form handling, validation, and CRUD operations.
- Apply authentication, role-based authorization, and deploy applications using ASP.NET Identity and hosting tools.

Course Learning Outcomes (CLOs):

By the end of the course, students will be able to:

1. Describe the MVC architecture and configure an ASP.NET MVC application.
2. Create interactive views, manage controllers, and handle application state.
3. Design data-driven MVC applications using Entity Framework with form validation.
4. Implement secure authentication, authorization, and deploy MVC applications.

Course Content

Module I: ASP.NET MVC Architecture and Application Setup (25%)

MVC Architecture Overview, ASP.NET MVC vs Web Forms, Setting up the development environment (Visual Studio, .NET SDK), Creating a basic MVC project, Routing and URL Mapping

Module II: Controllers, Views, and State Management (25%)

Controllers and Actions, Razor Syntax, Passing Data using ViewBag, ViewData, and TempData, Layouts and Partial Views, State Management: TempData, Session, Cookies

Module III: Models, Forms, and Entity Framework (25%)

Understanding Models in MVC, Introduction to Entity Framework, Code First Approach and Migrations, Creating Data Models and DbContext, Performing CRUD operations using EF, HTML & Tag Helpers, Model Binding and Form Validation using Data Annotations

Module IV: Authentication, Authorization, and Deployment (25%)

ASP.NET Identity Overview, Implementing Role-based Access, Middleware in MVC, Configuration: AppSettings & Environment Settings, Hosting & Deployment (IIS or local)

Text Books:

1. ASP.NET: The Complete Reference Books, McGraw Hill education

References Books:

1. Christian Nagel, "Professional C# .Net", Wrox Publication
2. Matthew Macdonald and Robert Standefer, "ASP.NET Complete Reference", TMH
3. Vijay Mukhi, "C# The Basics", BPB Publications

List of Open-Source Software/learning website:

1. www.c-sharpcorner.com
2. www.csharp-station.com/Tutorial.aspx
3. www.stackify.com/learn-asp-net-tutorials/

Practical List

1. Create a simple MVC application displaying static content using Controller and Razor View.
2. Use ViewBag, ViewData, and TempData to pass data between Controller and View.
3. Create a partial view for a common navigation bar and include it in multiple pages.
4. Create a layout page and apply it across all views for consistent UI.
5. Create a Student model with properties (ID, Name, Age, Email) and connect it using Entity Framework (Code First).
6. Perform full CRUD operations on the Student model.
7. Create and apply EF Migrations to update database schema changes.
8. Create a user registration form with built-in HTML helpers (@Html.TextBoxFor, @Html.LabelFor, etc.).
9. Implement client-side and server-side validations using Data Annotations.
10. Handle form submission errors and display validation messages using ModelState.
11. Display a list of records using a foreach loop and render as a table.
12. Add a search/filter box to filter displayed records dynamically.
13. Implement pagination using a custom logic or a library like PagedList.Mvc.
14. Create a login form with basic authentication (hardcoded users or Identity system).
15. Implement role-based access to restrict admin pages.
16. Use [Authorize] attribute to restrict controller access.
17. Create a feedback form that stores user data using TempData.
18. Implement a login system using session variables and cookies.
19. Track number of page visits using sessions.
20. Create a dropdown list bound to database data (e.g., country → city).
21. Use AJAX with jQuery to dynamically load partial views without refreshing the page.
22. Implement file upload functionality using MVC (upload profile pictures).
23. Generate and export a report in PDF/Excel format using Razor view.
24. Configure web.config for deployment and publish the site on IIS.
25. Create a web.config transformation for development vs production environments.
26. Setup a connection to SQL Server with secure credentials.
27. Implement logging using Serilog or NLog.

For CLO-PO mapping:

CLOs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CLO1	H	H	M	M	L	L	M	L	M	L	M	M
CLO2	H	H	H	M	M	L	M	M	H	M	L	M
CLO3	H	H	H	H	M	M	H	L	M	M	H	M
CLO4	H	H	M	M	L	M	M	L	M	L	M	L

For CLO-PSO mapping:

CLO	PSO1	PSO2	PSO3	PSO4
CLO1	H	M	L	M
CLO2	H	H	M	L
CLO3	H	M	H	M
CLO4	M	H	M	L
CLO5	H	H	L	H

Course: Virtualization & Applications of Cloud(Elective-2)								
Course Code:	Credit:4	Semester: III				me: MCA		
Teaching Scheme				Evaluation Scheme				
Lecture	Tutorial	Practical	Contact Hours	CIE Theory	CIE Practical	ESE Theory	ESE Practical	Total
3	0	2	5	25/50	-	25/50	-	50/100

Course Objectives:

- Demonstrate the system & software models and mechanisms that support cloud computing.
- Classify various cloud services and their providers
- Compare various cloud deployment models.
- Differentiate various types of computing environments and Identify enabling technologies of cloud computing.

Course Learning Outcomes (CLOs):

By the end of the course, students will be able to:

1. Students should grasp the characteristics, benefits, and various deployment models of cloud computing and understand the different service models (IaaS, PaaS, SaaS) and their implications
2. Students should be able to define and explain the purpose of virtualization, its different types (e.g., server, desktop, application), and its benefits and Students should be able to describe the key components and architecture of cloud platforms.
3. Students should understand the security challenges and best practices in the cloud environment.
4. Students should demonstrate practical skills in using virtualization tools and technologies to manage and optimize IT resources.

Course Content

Module I: Introduction to Cloud Computing (25%)

Eras of computing, The vision of Cloud Computing, Defining a cloud, A closer look, Cloud computing reference model, Historical developments, Different Stockholders in CC, Distributed systems, Virtualization, Web 2.0; Service oriented computing; Utility oriented computing.

Module II: Architectures for parallel and distributed computing (25%)

Parallel Vs Distributed computing, Elements of distributed computing, Technologies for distributed computing, What is Parallel Processing, Models for Inter-Process Communication, Remote Procedure call, Service Oriented Computing

Module III: Virtualization (25%)

Introduction, Characteristics of virtualized environments, Taxonomy of virtualization techniques, Virtualization and cloud computing, Pros and cons of virtualization, Technology, Types of Hardware Virtualization, examples: Xen: Para virtualization, VmWare: Full virtualization, Microsoft Hyper – V.

Module: IV- Cloud Computing Architecture, Tools and Applications (25%)

Introduction, Cloud reference model: Architecture, IaaS, PaaS, SaaS, Types of Clouds: Public,

Private, Hybrid and Community clouds, Economics of the cloud, Open challenges, Aneka PaaS; Open stack: Introduction to open stack; Components of open stack; Amazon web services; Google AppEngine; Microsoft Azure; Scientific applications: Healthcare; Biology; Geo-Science, Business and Consumer applications: ARM & ERP; Productivity; Social networking.

Text Books:

1. RjkumarBuyya, Christian Vecchiola, and ThamaraiSelci, Mastering Cloud Computing, Tata McGraw Hill, New Delhi, India, 2013.

References Books:

1. Cloud Computing for Dummies by Judith Hurwitz, R.Bloor, M. Kanfman, F.Halper (Wiley India Edition)
2. Cloud Computing: A Practical Approach by J.Vette, Toby J. Vette, Robert Elsenpeter (Tata McGraw Hill)
3. Kumar Saurabh, "Cloud Computing – insights into New -Era Infrastructure", Wiley India, 2011.
4. Toby Velte, Anthony Velte, Robert Elsenpeter, "Cloud Computing, A Practical Approach", TMH, 2009.

List of Suggestive s:

1. Create a word document of your class timetable and store locally and on the cloud with doc, and pdf Format (use www.zoho.com and docs.google.com)
2. Prepare a ppt on cloud computing –introduction, models, services, and Architecture ppt should contain explanations, images and at least 20 pages (use www.zoho.com and docs.google.com)
3. Create a google form of Employee Detail
4. Client server TCP based chatting application
5. Explain Hadoop Installation
6. HDFS Basic File Operations
7. HDFS Other commands
8. Map Reduce Word Count Example
9. Map Reduce Char Count Example
10. Case Study On Aws Web services.

PO-COMPETENCY-CLO MAPPING:

For CLO-PO mapping:

CLO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CLO1	M	M	M	M	M	M	M	M	M	M	M	M
CLO2	H	H	H	H	H	H	H	H	H	H	H	H
CLO3	M	M	M	M	M	M	M	M	M	M	M	M
CLO4	M	M	M	M	M	M	M	M	M	M	M	M

For CLO-PSO mapping:

CLO	PSO1	PSO2	PSO3	PSO4
CLO1	M	M	M	M
CLO2	H	H	H	H
CLO3	M	M	M	M
CLO4	M	H	H	M

Course: Big Data Tools (Elective-2)								
Course Code:	Credit:4	Semester: III				me: MCA		
Teaching Scheme				Evaluation Scheme				
Lecture	Tutorial	Practical	Contact Hours	CIE Theory	CIE Practical	ESE Theory	ESE Practical	Total
3	0	2	5	25/50	-	25/50	-	50/100

Course Objectives:

- Understand the concepts, characteristics, and challenges of Big Data.
- Gain hands-on knowledge of the Hadoop ecosystem and its components.
- Learn to process and analyze large datasets using MapReduce, Pig, and Hive.
- Explore NoSQL databases like HBase for managing unstructured data.
- Use tools like Sqoop, Flume, and Spark for data ingestion and real-time processing.

Course Learning Outcomes (CLOs):

By the end of the course, students will be able to:

1. Describe the fundamentals of Big Data, its architecture, and processing challenges.
2. Demonstrate the ability to configure and use Hadoop components such as HDFS and YARN.
3. Develop and execute data processing tasks using MapReduce, Pig, and Hive.
4. Apply NoSQL concepts and perform operations using HBase for scalable data storage.

Course Content

Module I: Introduction to Big Data and Hadoop Ecosystem (25%)

Introduction to Big Data: Definition, Characteristics (5 Vs), Use Cases, Traditional vs. Big Data Approaches, Hadoop Overview: History, Features, Architecture, HDFS (Hadoop Distributed File System): Concepts, Blocks, Replication, YARN (Yet Another Resource Negotiator): Architecture, Components, Installation and Configuration of Hadoop (Single-node and Multi-node)

Module II: Data Processing with MapReduce and Apache Pig (25%)

Introduction to MapReduce: Concepts, Architecture, and MapReduce Model, Writing MapReduce s in Java or Python, Word Count : InputSplit, Mapper, Reducer, Combiner, Limitations of MapReduce, Introduction to Apache Pig: Features, Architecture, Pig Latin Language: Data Types, Operators, Functions, Writing and Executing Pig Scripts (Batch and Interactive Modes)

Module III: Apache Hive and NoSQL Databases (25%)

Introduction to Hive: Architecture, HiveQL vs. SQL, Hive Metastore and Data Types, Creating Databases, Tables, and Performing CRUD Operations, Partitioning and Bucketing in Hive, Introduction to NoSQL: Characteristics, Types (Key-Value, Column, Document, Graph), Apache HBase: Architecture, Data Model, and Integration with Hadoop, HBase vs. Traditional RDBMS

Module IV: Data Ingestion, Processing, and Visualization Tools (25%)

Apache Sqoop: Import/Export between RDBMS and HDFS, Apache Flume: Data Collection and Aggregation, Apache Spark: Overview, RDDs, DataFrames, Spark SQL, Real-time Processing: Spark Streaming vs. Storm (Introduction Only), Data Visualization using tools like Tableau/Power BI (Basic Introduction), Case Studies: Big Data Applications in Industry (Finance, Healthcare, Social Media)

Text Books:

1. **"Big Data: Black Book" by DT Editorial Services** – Comprehensive guide covering the Hadoop ecosystem including HDFS, MapReduce, Hive, Pig, HBase, and Spark with practical examples suitable for beginners and professionals.
2. **"Hadoop: The Definitive Guide" by Tom White** – Authoritative reference on Hadoop architecture and mining, covering HDFS, MapReduce, YARN, and related tools with deep technical insights.
3. **"Big Data and Hadoop" by V.K. Jain** – Indian author-led textbook tailored to local university syllabi, featuring simplified explanations of Hadoop, Hive, Pig, and real-world applications.
4. **"Big Data Analytics" by Seema Acharya and Subhasini Chellappan** – Offers a blend of theoretical concepts and hands-on analytics using Hadoop, Hive, Pig, and advanced tools like Spark and Mahout.

References Books:

1. **"Mastering Big Data: Discovering Big Data Concepts, Components, and Tools" by Krish Krishnan** – Provides a comprehensive overview of Big Data architecture, technologies, and tools including Hadoop, NoSQL, and data integration strategies, suitable for both academic and professional learning.
2. **"Fundamentals of Big Data" by Saurabh Bhatia** – Student-friendly resource introducing core Big Data technologies and tools with focus on Hadoop components, NoSQL, and real-time processing.

List of Suggestive s:

1. Install and configure Hadoop in pseudo-distributed mode.
2. Perform basic file operations using HDFS commands.
3. Implement and execute a Word Count using MapReduce.
4. Analyze data using Apache Pig scripting language.
5. Create and query tables using Apache Hive.
6. Import data from MySQL into HDFS using Apache Sqoop.
7. Collect and store streaming data using Apache Flume.
8. Perform data operations using Apache HBase.
9. Process data using Apache Spark with RDD and DataFrame APIs.
10. Process real-time streaming data using Spark Streaming.

PO-COMPETENCY-CLO MAPPING:

For CLO-PO mapping:

CLO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CLO1	M	H	M	M	L	H	M	H	M	L	M	M
CLO2	H	M	H	H	H	M	H	L	H	M	H	H
CLO3	M	H	M	H	M	L	M	H	M	H	M	M
CLO4	M	H	H	H	H	M	M	L	H	M	M	H

For CLO-PSO mapping:

CLO	PSO1	PSO2	PSO3	PSO4
-----	------	------	------	------

CLO1	M	H	M	M
CLO2	H	L	M	H
CLO3	M	L	H	M
CLO4	M	H	H	M

Course: Digital Marketing with Content Writing (Elective-2)								
Course Code:		Credit:4		Semester: III			me: MCA	
Teaching Scheme				Evaluation Scheme				
Lecture	Tutorial	Practical	Contact Hours	CIE Theory	CIE Practical	ESE Theory	ESE Practical	Total
3	0	2	5	25/50	-	25/50	-	50/100

Course Objectives:

- The objective of Digital Marketing and Content Writing is to equip individuals with the skills to create compelling.
- The objective of Digital Marketing and Content Writing is to relevant content and utilize digital channels effectively to reach and engage target audiences.
- The objective of Digital Marketing and Content Writing is to ultimately driving brand awareness, customer acquisition, and business growth.
- The objective of Digital Marketing and Content Writing is to upon completing a digital marketing course, individuals will possess SEO, SEM, and Social Media Marketing expertise

Course Learning Outcomes (CLOs):

By the end of the course, students will be able to:

1. Expose and prepare students for a lucrative employment opportunity.
2. Equip students with demands of the digital world with global competency.
3. Bridge the gap between academia and industry.
4. The knowledge of various styles and techniques of writing and editing and the knowledge of new writing methods.

Course Content

Module I: Basics of Marketing (25%)

Marketing Environment, Marketing Plan, Perceptual Mapping, Consumer Behavior, Enhancing Customer Experience, Communicating with Consumers, Advertisements, Case Studies.

Module II: Basics of Digital Marketing (25%)

Evolution, Terminologies, Difference Between Traditional and Digital Marketing, Importance and Scope of Digital and Social Media Marketing, Online Retailing and Aggregators

Module III: Social Media Marketing (25%)

Defining Social Media Marketing Elements of Social Media Marketing, Social Media Vehicles, Elements of Social Media Marketing Strategies, Social Media Mix, Social Media Campaign Management, Social Media (Instagram, Linked in, Face book, etc)

Module: IV- Types of Content writing (25%)

Creative Creation (Hashtag, WhatsApp, Facebook, Instagram, YouTube, Twitter, Ads), Email Campaign Management Plagiarism laws in Content Writing What is plagiarism, rules on plagiarism, How to write plagiarism-free copies, The process of Content Writing – getting the brief, ideating, researching, structuring, formatting, Writing Styles - Non-fiction (Essays, Reports), Advertising, Newspapers, Corporate Communications -- Writing for business to business,(B2B), business to consumer (B2C), press releases, newsletters – focus on language, jargon, writing style, target audience, formal and informal language Visual Content Info graphics- Importance and relevance, Images, Screenshots, Videos, Memes, GIFs, 30 degree videos, Product Demonstrations Interactive

Content Quizzes, Polls, Interactive white papers

Text Books:

1. Feldar, Lynda is Writing for the Web: Creating Compelling Web Content Using Words, Pictures, and Sound.

References Books:

1. James, Anthony. Blog Writing : The Content Creation Blueprint.
2. Devanshi sharma by power of content writing.
3. Digital Marketing Third Edition by Seema Gupta
4. Buy. Logy by Martin Lindstrom

List of Suggestive s:

1. Create a Facebook Business Page
2. Write a Blog Post (150 words)
3. Create an Instagram Business Profile
4. Keyword Research Using Google
5. Write an Email Newsletter
6. Create a Simple Google Ad Copy
7. Write Social Media Posts
8. SEO (Search Engine Optimization)
9. Analytics & Reporting
10. Affiliate & Influencer Marketing

PO-COMPETENCY-CLO MAPPING:

For CLO-PO mapping:

CLO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CLO1	M	M	M	M	M	M	M	M	M	M	M	M
CLO2	H	H	H	H	H	H	H	H	H	H	H	H
CLO3	M	M	M	M	M	M	M	M	M	M	M	M
CLO4	M	M	M	M	M	M	M	M	M	M	M	M

For CLO-PSO mapping:

CLO	PSO1	PSO2	PSO3	PSO4
CLO1	M	M	M	M
CLO2	H	H	H	H
CLO3	M	M	M	M
CLO4	M	M	M	M

