



**Maharashtra Education Society's
Abasaheb Garware College
(Autonomous)**

(Affiliated to Savitribai Phule Pune University)

**Three Year B.Sc. Degree Program in Computer Science
(Faculty of Science and Technology)**

T.Y.B.Sc. (Computer Science)

**Choice Based Credit System Syllabus
Syllabi under NEP
To be implemented from Academic Year 2026-2027**

Course Structure of B.C.A (Science)
Fourth Year

Year	Semester	Course Type	Course Code	Course Title	Credit	No. of hours
1	I	Core	CA-401-MJ	Advanced Web Technologies	4	48
			CA-402-MJ	Network Security and Cryptography	4	48
			CA-403-MJ	Data Structures and Algorithms	4	48
			CA-404-MJP	Practical based on Advanced Web Technologies, Data Structures and Algorithms	2	4 per batch
		Elective	CA-410-MJ	Cloud Application Development	2	30
			CA-411-MJP	Lab course based on Cloud Application Development	2	4 per batch
			OR			
			CA-412-MJ	Advanced Python Programming	2	30
			CA-413-MJP	Lab course based on Advanced Python Programming	2	4 per batch
			CA-440-RM	Research Methodology	4	48
	II	Core	CA-451-MJ	Full Stack Development - I	4	48
			CA-455-MJ	Software Architecture and Design Patterns	4	48
			CA-453-MJ	Machine Learning	4	48
			CA-454-MJP	Practical based on Full Stack Development – I and Machine Learning	2	4 per batch
		Elective	CA-460-MJ	Advanced Database	2	30
			CA-461-MJP	Lab based on Advanced Database	2	4 per batch
			OR			
			CA-462-MJ	.NET	2	30
			CA-463-MJP	Lab based on .NET	2	4 per batch
			CA-481-OJT	On Job Training	4	-

CA-401-MJ: Advanced Web Techniques

Lectures: 48 (Credits-4)

Prerequisites:

- Basic knowledge of HTML, CSS, JavaScript
- Fundamentals of Programming
- Basic understanding of Web Technologies

Learning Objectives:

1. To revise core web technologies required for modern web development.
2. To understand PHP fundamentals and object-oriented concepts.
3. To develop dynamic web applications using Laravel framework.
4. To implement database operations using Eloquent ORM.
5. To design secure web applications with authentication and REST APIs.

Course Outcomes: On completion of this course, students will be able to:

- CO1: Apply web fundamentals and PHP concepts in application development.
CO2: Understand MVC architecture and develop applications using Laravel.
CO3: Implement routing, controllers, and middleware in Laravel.
CO4: Design dynamic views and perform form handling with validation.
CO5: Develop database-driven, secure web applications with authentication and REST APIs and basic advanced Laravel features.

Unit 1: Web Foundations & PHP Basics

08 (C01)

- 1.1 Web Fundamentals: Evolution of Web, Client-Server Architecture, HTTP/HTTPS, FTP, IP, HTTP Request-Response cycle
- 1.2 HTML5 (Revision): Structure of HTML document, Semantic tags, Forms and input types
- 1.3 CSS (Revision): CSS types, Basic styling, Layout (flexbox, display)
- 1.4 JavaScript (Revision): Variables, operators, control structures, Functions and events, DOM manipulation, Form validation
- 1.5 PHP Basics: Syntax, variables, data types, Control structures, Functions, Arrays
- 1.6 Object-Oriented PHP: Classes and objects, Constructors, Inheritance, Interfaces

Unit 2: Introduction to Laravel & MVC

06 (C02)

- 2.1 Introduction to Laravel: Features and advantages
- 2.2 MVC Architecture: Model, View, Controller concept, MVC in Laravel
- 2.3 Installation and Setup: Composer installation, Laravel project setup
- 2.4 Application Structure: Root, app, routes, resources
- 2.5 Configuration: .env file, Database configuration
- 2.6 Artisan CLI: Basic commands

Unit 3: Routing, Controllers & Middleware

08 (C03)

- 3.1 Routing: Basic routing, Route parameters, Named routes
- 3.2 Controllers: Creating controllers, Resource controllers
- 3.3 Middleware: Concept and usage, Route protection
- 3.4 Dependency Injection: Constructor & method injection

Unit 4: Views, Blade & Forms	06 (C04)
4.1 Views: Creating and managing views, Passing data	
4.2 Blade Templating: Syntax and layouts, Components	
4.3 Forms: Form handling, CSRF protection	
4.4 Validation: Server-side validation, Error handling	
Unit 5: Database & Eloquent ORM	08 (C05)
5.1 Database Setup: Configuration	
5.2 Migrations: Create, run, modify	
5.3 Eloquent ORM: Models, CRUD operations	
5.4 Relationships: One-to-One, One-to-Many, Many-to-Many	
Unit 6: Authentication, Security & REST API	08 (C05)
6.1 Authentication: Login & Registration: Authentication scaffolding	
6.2 Middleware for Auth: Protecting routes	
6.3 Security: Password hashing, CSRF protection, Best practices	
6.4 REST API Development: API routes, JSON responses	
Unit 7: Advanced Laravel	04 (C05)
7.1 Queues and Jobs	
7.2 Events and Listeners	
7.3 Task Scheduling	
7.4 File Upload and Storage	
7.5 Caching strategy	

Reference Books:

1. Laravel: Up & Running, *A Framework for Building Modern PHP Apps*, Matt Stauffer, O'Reilly Publication.
2. HTML & CSS: The Complete Reference, Fifth Edition, McGraw Hill.
3. Eloquent JavaScript, No Starch Press.
4. PHP and MySQL Web Development, Addison-Wesley.
5. Programming PHP, O'Reilly Publication.
6. Web Technologies Black Book, Dreamtech Press.

Online Resources and Documentation:

1. Laravel Official Documentation: <https://laravel.com/docs>
2. PHP Official Manual: <https://www.php.net/docs.php>
3. Mozilla Developer Network (HTML, CSS, JavaScript): <https://developer.mozilla.org>

NPTEL / SWAYAM Resources:

1. NPTEL: <https://nptel.ac.in>
2. SWAYAM: <https://swayam.gov.in>

CA-402-MJ: Network Security and Cryptography

Lectures:48 (Credits-4)

Prerequisites:

- Networking Protocols & Models (OSI & TCP/IP)
- TCP, UDP, HTTP/HTTPS
- IP Addressing (IPv4, IPv6), Subnetting
- Basics of Firewalls and VPNs

Learning Objectives:

1. To understand security principles and cryptographic foundations.
2. To analyze symmetric and asymmetric cryptographic algorithms.
3. To study hashing, digital signatures, and emerging cryptographic trends.
4. To explore secure communication protocols and PKI.
5. To evaluate network security mechanisms and defense strategies.

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

- C01: Explain security concepts, threats, and cryptographic fundamentals.
C02: Analyze and apply symmetric key cryptographic techniques.
C03: Implement asymmetric cryptography, hashing, and digital signatures.
C04: Apply secure communication protocols and certificate-based security.
C05: Evaluate network security mechanisms and defense systems.

Unit 1: Security Fundamentals & Classical Cryptography **08 (C01)**

- 1.1 Security Principles and Need for Security
- 1.2 CIA Triad and Types of Attacks (Passive & Active)
- 1.3 Fundamentals of Cryptography: Plaintext, Ciphertext, Encryption, Decryption
- 1.4 Cryptanalysis and Brute Force Attacks
- 1.5 Classical Techniques: Substitution and Transposition
- 1.6 Key Space, Key Size and Security Strength
- 1.7 Steganography: Techniques and Applications

Unit 2: Symmetric Key Cryptography **10 (C02)**

- 2.1 Introduction to Symmetric Key Cryptography
- 2.2 Block Cipher and Stream Cipher
- 2.3 Modes of Operation: ECB, CBC, CFB, OFB
- 2.4 Data Encryption Standard (DES) and Variants
- 2.5 Advanced Encryption Standard (AES)
- 2.6 Key Distribution Problem and Security Considerations

Unit 3: Asymmetric Cryptography & Hashing **10 (C03)**

- 3.1 Introduction to Asymmetric Key Cryptography
- 3.2 Symmetric vs Asymmetric Cryptography
- 3.3 RSA Algorithm: Key Generation, Encryption and Decryption
- 3.4 Diffie-Hellman Key Exchange

- 3.5 Digital Signatures: Concept and Implementation
- 3.6 Hash Functions and Properties
- 3.7 Message Digests: MD5 and SHA Family (Overview)
- 3.8 Introduction to Post-Quantum Cryptography

Unit 4: Internet Security Protocols & PKI

08 (C04)

- 4.1 Digital Certificates: CA, Structure, Creation, Types, Revocation
- 4.2 Public Key Infrastructure (PKI): Concepts and Components
- 4.3 SSL and TLS Protocols
- 4.4 HTTPS and Secure Web Communication
- 4.5 Secure HTTP (SHTTP)
- 4.6 Time Stamping Protocol (TSP)

Unit 5: Network Security Mechanisms

06 (C05)

- 5.1 Firewalls: Types and Architecture
- 5.2 Intrusion Detection and Prevention Systems (IDS/IPS)
- 5.3 Authentication Mechanisms: Password-based, Token-based, Biometric Authentication

Unit 6: VPN, IPSec & Advanced Security Concepts

06 (C05)

- 6.1 Virtual Private Networks (VPNs): Concepts and Types
- 6.2 IPSec: Architecture and Protocols
- 6.3 Secure Network Design Principles
- 6.4 Introduction to Zero Trust Security

Reference Books:

- 1. Cryptography and Network Security – Atul Kahate, 4th Edition, McGraw Hill Publication.
- 2. Cryptography and Network Security – William Stallings, 8th Edition, Pearson Publication.
- 3. Network Security Essentials: Applications and Standards – William Stallings, 6th Edition, Pearson Publication.
- 4. Cryptography: Theory and Practice – Douglas Stinson, CRC Press.

CA-403-MJ: Data Structures and Algorithms

Lectures:48 (Credits-4)

Prerequisites:

- Sound knowledge of algorithms and any programming language

Learning Objectives:

1. To design the algorithms
2. To select the appropriate algorithm by doing necessary analysis of algorithms
3. To learn basic Algorithm Analysis techniques and understand the use of asymptotic notations
4. Understand different design strategies

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

- C01: Choose appropriate data structures depending upon the nature of the problem.
- C02: Apply suitable algorithm design strategy to create efficient solutions for solving complex problems.
- C03: Find optimal solutions for solving problems.

Unit 1: Basics of Algorithms

10 (C02, C03)

- 1.1 Algorithm definition and characteristics
- 1.2 Characterizing Run Time, Use of asymptotic notation, Big-Oh, Big Omega, Big Theta Notations
- 1.3 Space complexity
- 1.4 Analyzing Iterative and Recursive Algorithms, Recurrence relations
- 1.5 Divide and Conquer strategy, Sorting Problem (Merge Sort, Quick Sort)
- 1.6 Sorting in linear time: counting sort, concept of bucket sort
- 1.7 Searching algorithm: Binary with time complexity

Unit 2: Data Structures

12 (C01, C02)

- 2.1 Introduction, Linear and nonlinear data structures
- 2.2 Linear data structures: Linked Lists, Stacks, Queues Concept and applications
- 2.3 Non Linear data structures
 - 2.3.1 Tree - Concept of tree and Terminologies
 - 2.3.2 Concept and types of Binary trees - skewed tree, strictly binary tree, full binary tree, complete binary tree, expression tree, binary search tree, Heap tree
 - 2.3.3 Representation of binary tree (dynamic)
 - 2.3.4 Traversal methods of Binary tree – preorder, inorder, postorder
 - 2.3.5 Operations on binary search tree - create, insert node, search a value
 - 2.3.6 Applications of tree
 - 2.3.7 Graph - Concept of graph and terminologies
 - 2.3.8 Representation of Graph – Adjacency matrix, Adjacency list
 - 2.3.9 Graph Traversals – Breadth First Search and Depth First Search
 - 2.3.10 Applications of graph

Unit 3: Greedy method**08 (C02, C03)**

- 3.1 Control Abstraction
- 3.2 Fractional Knapsack Problem
- 3.3 Optimal merge patterns
- 3.4 Huffman coding
- 3.5 Minimum Spanning Tree: Kruskal's and Prim's algorithm

Unit 4: Dynamic Programming**12 (C02, C03)**

- 4.1 Principle of optimality
- 4.2 0/1 Knapsack Problem (function method)
- 4.3 Matrix chain multiplication
- 4.4 All-pairs Shortest Path Problem (Floyd-Warshall's algorithm)
- 4.5 Bellman-Ford Algorithm
- 4.6 Longest Common Subsequence
- 4.7 String Editing
- 4.8 Traveling Salesperson Problem

Unit 5: Decrease and Conquer**06 (C02, C03)**

- 5.1 Topological sorting
- 5.2 Connected components and spanning trees
- 5.3 Articulation Point and Bridge edge

Reference Books:

1. Fundamentals of Data Structures, Horowitz and Sahani, Galgotia Publications.
2. Fundamentals of Computer Algorithms, Horowitz and Sahani, Galgotia Publications.
3. Introduction to Algorithms, TH Cormen, CE Leiserson, RL Rivest, C Stein, PHI.

CA-404-MJ: Lab based on Advanced Web Technologies, Data Structures and Algorithms

No. of Sessions: 12 (Credits-2)

List of Advanced Web Technologies Assignments

Assignment 1: Web Design using HTML, CSS & JavaScript

1. Create a form with using HTML5 tags.
2. Apply CSS for styling and layout (flexbox, display properties).
3. Implement JavaScript for: Event handling, DOM manipulation (content and style changes)
4. Perform client-side form validation using JavaScript.

Assignment 2: Server-side Programming using PHP

1. Write PHP scripts using variables, data types, control structures, and functions.
2. Perform string and array operations using built-in PHP functions.
3. Develop a PHP program demonstrating Object-Oriented Programming concepts.

Assignment 3: Introduction to Laravel & MVC

1. Install Laravel using Composer and create a new project.
2. Explore project structure and configure .env file.
3. Create a simple MVC-based application displaying static pages.

Assignment 4 Routing, Controllers & Middleware

1. Implement routing with parameters and named routes.
2. Create controllers and resource controllers.
3. Implement middleware for route protection.

Assignment 5: Views, Blade & Forms

1. Create views and pass data from controller.
2. Design layouts using Blade templates.
3. Implement form handling with validation and error messages.

Assignment 6: Database & Eloquent ORM

1. Configure database and create migrations.
2. Perform CRUD operations using Eloquent ORM.
3. Implement relationships (One-to-One, One-to-Many, Many-to-Many).

Assignment 7: Authentication & REST API

1. Implement login and registration system.
2. Secure routes using authentication middleware.
3. Develop REST APIs returning JSON responses.
4. Implement file upload functionality.

List of Data Structures and Algorithms Assignments

Student can do assignment using C programming language.

Assignment 1: Implement linear data structure - Linked list

Assignment 2: Stack and Queue

Assignment 3: Tree and Graphs

Assignment 4: Divide and Conquer & Sorting

Implement merge sort and quick sort.

Assignment 5: Implementation of Shortest path algorithms

CA-410-MJ: Cloud Application Development

Lectures: 30 (Credits-2)

Prerequisites:

- Cloud Computing

Learning Objectives:

1. To understand principles of cloud-native applications.
2. To learn cloud development tools.
3. To develop and deploy cloud applications using CI/CD pipelines.
4. To identify major cloud service providers and their offerings.
5. To understand the design of security architecture in cloud computing.

Course Outcomes: On completion of this course, students will be able to:

CO1: Explain cloud-native architectures, application design principles, and service models.

CO2: Apply cloud development tools, containers, and orchestration technologies.

CO3: Design and deploy cloud applications using CI/CD and IaC practices.

CO4: Analyze and implement cloud data management solutions.

CO5: Apply security, scaling, and optimization techniques in cloud environments.

Unit 1: Cloud Application Architecture & Design

06 (CO1)

- 1.1 Principles of Cloud-Native Applications
- 1.2 Monolithic vs Microservices Architecture
- 1.3 12-Factor App Methodology
- 1.4 RESTful APIs and API Design
- 1.5 Introduction to Serverless Architecture

Unit 2: Development Tools & Platforms

06 (CO2)

- 2.1 Cloud Development Environments (IDE, CLI tools)
- 2.2 SDKs and APIs of Cloud Platforms
- 2.3 Introduction to Containers and Docker
- 2.4 Basics of Kubernetes for Application Deployment

Unit 3: Building and Deploying Cloud Applications

07 (CO4)

- 3.1 Application Deployment Models in Cloud
- 3.2 Continuous Integration & Continuous Deployment (CI/CD)
- 3.3 Infrastructure as Code (IaC) – Basics
- 3.4 Deploying Applications on Cloud Platforms
- 3.5 Monitoring and Logging of Applications

Unit 4: Data Management in Cloud Applications

06 (CO5)

- 4.1 Cloud Databases – SQL vs NoSQL
- 4.2 Data Storage Services in Cloud
- 4.3 Data Integration and APIs
- 4.4 Caching Techniques (Redis, CDN basics)
- 4.5 Backup and Disaster Recovery

Unit 5: Security, Scaling & Optimization**05 (C05)**

- 5.1 Authentication and Authorization in Cloud Apps
- 5.2 Secure API Development
- 5.3 Auto Scaling and Load Balancing in Applications
- 5.4 Cost Optimization Techniques
- 5.5 Performance Optimization and Best Practices

Reference Books

1. Cloud Computing: Principles and Paradigms, Editors, Rajkumar Buyya, James Broberg, Andrzej M. Goscinski, Wiley, 2011.
2. Enterprise Cloud Computing - Technology, Architecture, Applications, Gautam Shroff, Cambridge University Press, 2010.
3. Cloud Security: A Comprehensive Guide to Secure Cloud Computing, Ronald L. Krutz, Russell Dean Vines, Wiley- India, 2010.
4. AWS, The ultimate guide from beginners to advanced, Maveric Koston.
5. Microsoft Azure: Planning, Deploying, and Managing Your Data Center in the Cloud, Anthony Puca, Mike Manning, Marshal Copeland, Julian Soh, David Gollob.

CA-411-MJ: Lab based on Cloud Application Development

No of Sessions: 12 (Credits-2)

Assignments on Cloud Application Development:

1. Cloud-Native Application
2. Containerization using Docker
3. Kubernetes Deployment Basics
4. CI/CD Pipeline Setup
5. Implement authentication and authorization

CA-412-MJ: Advanced Python Programming

Lectures: 30 (Credits-2)

Prerequisites:

- Basic Python Programming concepts

Learning Objectives:

1. Understand OOP concepts in Python.
2. Learn database connectivity and CRUD operations.
3. Explore web scraping using BeautifulSoup.
4. Understand Flask-based web development.
5. Develop simple web apps and APIs.

Course Outcomes: On completion of this course, students will be able to

- CO1: Apply OOP principles in Python programs.
- CO2: Build database-driven applications.
- CO3: Implement web scraping and API-based data extraction using modern libraries.
- CO4: Process structured and semi-structured data using Python file systems and regex.
- CO5: Develop web applications and REST APIs using Flask.

Unit 1: Object Oriented Programming in Python

08 (CO1)

- 1.1 Concept of Object Oriented Programming
- 1.2 Classes and Objects
- 1.3 Constructor (__init__ method)
- 1.4 Instance variables and class variables
- 1.5 Inheritance types (Single inheritance, Multiple inheritance, Multilevel inheritance)
- 1.6 Method Overriding
- 1.7 Special methods (__str__, __add__, etc.)

Unit 2: Database Connectivity

06 (CO2)

- 2.1 Introduction to Database Programming
- 2.2 SQLite / MySQL connectivity with Python
- 2.3 CRUD operations (Insert, Update, Delete, Select)
- 2.4 Database driven applications

Unit 3: Python for Web Development

10 (CO3, CO5)

- 3.1 Introduction to Web Applications and Client-Server Architecture
- 3.2 Overview and setup of Flask
- 3.3 Creating and running Flask applications
- 3.4 URL routing and handling HTTP requests
- 3.5 HTML templates using Jinja
- 3.6 Handling forms and user input
- 3.7 Building simple REST APIs in Flask
- 3.8 Web Scrapping with BeautifulSoup and Scrapy

Unit 4: File handling and regular expressions

06 (CO4)

- 4.1 Working with text, CSV, JSON, and XML files
- 4.2 File I/O operations and context managers
- 4.3 Introduction to regular expressions(regex)
- 4.4 Pattern matching and string manipulations
- 4.5 Advance regex techniques in Python

Reference Books:

1. Learning Python By Mark Lutz, 5th Edition, O'Reilly Media, ISBN 978-1-449-35573-9
2. Python Crash Course By Eric Matthes, 2nd Edition, No Starch Press, ISBN 978-1-59327-928-8
3. Flask Web Development By Miguel Grinberg, 2nd Edition, O'Reilly Media, ISBN 978-1-491-99173-2
4. Web Scraping with Python By Ryan Mitchell, 2nd Edition, O'Reilly Media, ISBN 978-1-491-98219-8
5. Python Database Programming Cookbook By Rick Copeland, 2nd Edition, Packt Publishing, ISBN 978-1-782-16700-9

CA-413-MJP: Laboratory on Advanced Python Programming

No. of Sessions: 12 (Credits-2)

Learning Objectives:

1. Apply OOP concepts to real-world problems.
2. Develop database-driven applications using Python.
3. Perform web scraping and API-based data extraction.
4. Build web applications using Flask.
5. Work with structured data formats like CSV and JSON.
6. Develop deployable mini web applications.

List of Experiments

1. Object Oriented Programming in Python

- a. Write a Python program to create a class Student with attributes and methods to display student details.
- b. Implement constructor (init) and demonstrate instance and class variables.
- c. Write a program demonstrating single inheritance.
- d. Write a program demonstrating multiple inheritance.
- e. Write a program demonstrating multilevel inheritance.
- f. Implement method overriding with base and derived classes.

2. Database Connectivity

- a. Create database table using Python, Insert records into database.
- b. Perform CRUD operations (Insert, Update, Delete, Select).
- c. Develop Student Database Application with menu-driven interface.
Case study: Create a Login System using database.

3. Python for Web Development (Flask)

- a. Install Flask and create first Flask application.
- b. Implement URL routing in Flask.
- c. Create HTML templates using Jinja.
- d. Create Flask form to collect user input.
- e. Store form data into database using Flask.
- f. Develop simple REST API using Flask.
- g. Implement Web Scraping using BeautifulSoup.
- h. Scrape product/job/news data from website and store into CSV.

4. File Handling and Regular Expression

- a. Read and write text files.
- b. Read and write CSV files
- c. Read and write JSON file
- d. Parse XML files using Python.
- e. Use regex to validate email addresses
- f. Use regex to extract phone numbers from text.

Reference Links:

1. <https://www.programiz.com/python-programming/object-oriented-programming>
2. <https://www.sqlitetutorial.net/sqlite-python/>
3. <https://flask.palletsprojects.com/en/latest/quickstart/>
4. <https://realpython.com/beautiful-soup-web-scraper-python/>
5. <https://realpython.com/python-csv/>
6. <https://regex101.com/>

CA-440-MJ: Research Methodology

Lectures: 48 (Credits- 04)

Learning Objectives:

1. Research Methodology course is designed to equip students with the necessary knowledge, skills, and understanding of various research techniques and methodologies.
2. Students should be familiar with various data collection techniques, such as surveys, interviews, observations, and experiments, and understand their strengths and limitations.
3. Students should be aware of ethical considerations in research, including issues related to participant consent, privacy, confidentiality, and avoiding plagiarism.
4. Its aim is to enable students to conduct research effectively, critically evaluate existing research, and contribute to the advancement of knowledge in their respective fields.

Course Outcomes: On completion of this course, students will be able to

- C01: Understand of the fundamental concepts of research, including the research process, research questions, hypotheses, and variables.
- C02: Conduct a comprehensive literature review to identify relevant studies, synthesize existing knowledge, and identify research gaps.
- C03: Identify research problems, formulate research questions, and design appropriate methodologies to address these problems
- C04: Identify and select appropriate research designs, such as experimental, observational, survey, qualitative, or mixed-methods, based on the research objectives.
- C05: Apply appropriate data analysis methods, including statistical techniques or qualitative analysis, to draw meaningful conclusions from research data.
- C06: Develop a well-structured research proposal, outlining research questions, methodology, expected outcomes, and a rationale for the study.
- C07: Communicate research findings effectively through written reports, presentations, and academic papers.
- C08: Gain an appreciation for the importance of research in contributing to the advancement of knowledge in their field of study and broader society.
- C09: Understand the principles of research ethics and integrity and apply them in their research.

Unit 1: Introduction to Research Methodology

08 (C01, C09)

- 1.1 Meaning of Research
- 1.2 Objectives of Research
- 1.3 Types of Research
- 1.4 Research Approaches
- 1.5 Significance of Research
- 1.6 Research Ethics and Integrity
- 1.7 Plagiarism and types of plagiarism
- 1.8 Research Methods versus Methodology
- 1.9 Criteria of Good Research

Unit 2: Literature Review and Formulation of Research Problems

06 (C01, C02, C03)

- 2.1 Research Process

- 2.2 Reviewing the literature: purpose of a literature review
- 2.3 Literature resources
- 2.4 The Internet and a literature review
- 2.5 Conducting and Evaluating literature reviews
- 2.6 Formulation of research problem
 - 2.6.1 What is a Research Problem?
 - 2.6.2 Selecting the Problem
 - 2.6.3 Necessity of Defining the Problem
 - 2.6.4 Technique Involved in Defining a Problem

Unit 3: Research Design

06 (C03, C04)

- 3.1 Meaning of Research Design
- 3.2 Need for Research Design
- 3.3 Features of a Good Design
- 3.4 Important Concepts Relating to Research Design
- 3.5 Different Research Designs/Methods
 - 3.5.1 Pure and Applied Research
 - 3.5.2 Exploratory or Formulative Research
 - 3.5.3 Descriptive Research
 - 3.5.4 Diagnostic Research
 - 3.5.5 Evaluation Studies
 - 3.5.6 Action Research
 - 3.5.7 Experimental Research
 - 3.5.8 Analytical Study or Statistical Method
 - 3.5.9 Historical Research
 - 3.5.10 Surveys
 - 3.5.11 Case Study
 - 3.5.12 Field Studies

Unit 4: Hypothesis and Sampling

06 (C05, C06)

- 4.1 What is Hypothesis?
- 4.2 Nature & Characteristics of Hypothesis
- 4.3 Significance of Hypothesis
- 4.4 Types of Hypothesis
- 4.5 Sources of Hypothesis
- 4.6 Characteristics of Good Hypothesis
- 4.7 What is sampling?
- 4.8 Characteristics of Good Sample
- 4.9 Basis of Sampling
- 4.10 Merits and demerits of Sampling
- 4.11 Sampling Techniques or Methods
- 4.12 Sample Design and Choice of Sampling Technique

Unit 5: Data Collection, Processing and Analysis of Data

06 (C05)

- 5.1 Collection of Primary Data
- 5.2 Method of data Collections - Observation, Interview, Questionnaires and Schedules

- 5.3 Difference between Questionnaires and Schedules
- 5.4 Some Other Methods of Data Collection
- 5.5 Collection of Secondary Data
- 5.6 Selection of Appropriate Method for Data Collection
- 5.7 Case Study Method
- 5.8 Processing Operations and Some Problems in Processing
- 5.9 Elements/Types of Data Analysis
- 5.10 Measures of Central Tendency, Dispersion, Asymmetry (Skewness)
- 5.11 Measures of Relationship - Chi-Square, t-test, ANNOVA(f-test), Z-test
- 5.12 Simple Regression Analysis, and Multiple Correlation and Regression
- 5.13 Quantitative and Qualitative Data Analysis Tools

Unit 6: Interpretation and Report Writing

08 (C06, C07, C08)

- 6.1 Meaning of Interpretation, Why Interpretation?
- 6.2 Technique of Interpretation
- 6.3 Precaution in Interpretation
- 6.4 Significance of Report Writing
- 6.5 Different Steps in Writing Report
- 6.6 Layout of the Research Report
- 6.7 Types of Reports (Research Proposal/Synopsis, Research Paper, and Thesis)
- 6.8 Oral Presentation
- 6.9 Mechanics of Writing a Research Report
- 6.10 Precautions for Writing Research Reports

Unit 7: Publication Ethics and Open Access Publishing

08 (C07, C09)

- 7.1 Publication ethics: definition, introduction and importance
 - 7.2 Best practices/standards setting initiatives and guidelines: COPE, WAME, etc.
 - 7.3 Conflicts of interest
 - 7.4 Publication misconduct: definition, concept, problems that lead to unethical behaviour and vice versa, types
 - 7.5 Violation of publication ethics, authorship and contributor ship
 - 7.6 Identification of publication misconduct, complaints and appeals
 - 7.7 Predatory publishers and journal
 - 7.8 Open access publications and initiatives
 - 7.9 SHERPA/RoMEO online resource to check publisher copyright & self-archiving policies
 - 7.10 Software tool to identify predatory publications developed by SPPU
 - 7.11 Journal finder/ journal suggestion tools viz. JANE, Elsevier Journal Finder, Springer Journal Suggester, etc.
- E-Resources for research: GoogleScholar, ShodhGanaga, ShodhGangotri, SciHub, etc...

Reference Books:

1. Researching Information Systems and Computing by Briony J Oates, SAGE SOUTH ASIA EDITION
2. Research Methodology: A Step-by-Step Guide for Beginners, Kumar, Pearson Education.
3. Research Methodology Methods and Techniques, Kothari, C. R., Wiley Eastern Ltd.
4. The Research Methods Knowledge Base, by William M. K. Trochim, James P. Donnelly

5. Introducing Research Methodology: A Beginner's Guide to Doing a Research Project, by Uwe Flick
6. A Guide to Research and Publication Ethics by Partha Pratim Ray, New Delhi Publishers
7. RESEARCH & PUBLICATION ETHICS by Wakil Kumar Yadav, NOTION PRESS
8. Practical Research Methods, Dawson, C., UBSPD Pvt. Ltd.

CA-451-MJ: Full Stack Development - I

Lectures: 48 (Credits- 04)

Prerequisites:

- Basics of Scripting Language
- Concept of client and server
- Concept of Web Server

Learning Objectives:

1. To understand and learn modern web technologies.
2. To understand server-side programming.
3. To learn developing basic interactive web applications.
4. To learn basic level Full Stack Development Course
5. To learn NOSQL database

Course Outcomes: On the completion of this course students will be -

- C01: Able to write programs that are server-side scripting.
- C02: Understand the client-server model of the web
- C03: Familiar with latest web development tools and techniques in IT industry.
- C04: Ready for advanced level of Full Stack Development Course.
- C05: Familiar with NoSQL database and ready learn for advanced part for database

Unit 1: Introduction

04 (C01)

- 1.1 How Web works?
- 1.2 What is Full Stack Development?
- 1.3 What is ECMA Script?
- 1.4 Concept of Client-Side Scripting
- 1.5 Concept of Server-Side Scripting
- 1.6 Concept of Framework

Unit 2: Basics of JavaScript

10 (C02)

- 2.1 JavaScript Variables and Operators
- 2.2 JavaScript Arrays and Functions
- 2.3 Importance and need of JavaScript Objects
- 2.4 JavaScript Events
- 2.5 Async Functions
- 2.6 Promises and Callback

Unit 3: Introduction to Node.js and NPM

12 (C03)

- 3.1 Traditional Web Server Model
- 3.2 Concept of Node.js and Event Loop
- 3.3 Installing Node.js
- 3.4 Components of Node.js Application
- 3.5 Modules, Types of Modules
- 3.6 Concept and examples of export
- 3.7 Create a basic application
- 3.8 What is Node Package Manager (NPM)

- 3.9 Packages in Node.js
- 3.10 Local and global Packages
- 3.11 Dependencies in packages

Unit 4: Express.js

12 (C04)

- 4.1 REST, RESTful services
- 4.2 Introduction to Express Frameworks
- 4.3 Routing
- 4.4 Routes, Views and Template Engine
- 4.5 Handling HTTP GET/POST request
- 4.6 Calling Endpoints using Postman
- 4.7 Using middleware and types of middleware
- 4.8 Form Fetching Data
- 4.9 Form Validation

Unit 5: Working with Databases

10 (C05)

- 5.1 Connection String
- 5.2 Configuring
- 5.3 Working with Select command
- 5.4 Various database operations
- 5.5 MongoDB
- 5.6 Mongoose ODM
- 5.7 Mongoose Schema
- 5.8 Mongoose Model
- 5.9 Querying with Mongoose
- 5.10 Aggregations in mongodb

Reference Books:

- 9. Mastering Node.js, Sandro Pasquali, Packt
- 10. Node.js complete reference guide, Velentin Bojinov, David Herron, Dioge Resende, Packt
- 11. Smashing Node.js, JavaScript Everywhere, Guillermo Rauch, WILEY

CS-455-MJ Software Architecture and Design Patterns

Lectures: 48 (Credits 4)

Pre-requisites:

- Familiarity with UML and OOPs Concepts
- Programming in Java

Learning Objectives:

1. To introduce students to the basic concepts and techniques of SADP.
2. Recognize the characteristics of patterns that make it useful to solve real-world problems.
3. To understand the use of specific frameworks as per applications need.
4. Design java application using design pattern techniques.

Course Outcomes: On completion of this course, students will be able to:

- CO1: Understand basic concepts and techniques of SADP.
- CO2: Understand different architectural styles.
- CO3: Able to use of patterns and architectures for solving practical problems.
- CO4: Understand about design pattern and write java programs using Design Pattern.
- CO5: Understand about GRASP.
- CO6: Understand different frameworks and web architectures.

Unit 1: Introduction

02 (C01)

- 1.1 UML The Notation
- 1.2 Process Unified Process / Rational Unified Process inception, elaboration, construction, transition
- 1.3 How various components fit in the life cycle, the artifacts at end of each process / discipline

Unit 2: Software Architecture

03 (C01)

- 2.1 Introduction to Software Architecture
- 2.2 Importance of architecture
- 2.3 Architectural structures and views

Unit 3: Architectural Styles

05 (C02)

- 3.1 Architectural Styles
- 3.2 Pipes and Filters
- 3.3 Data Abstraction and Object – Oriented Organization
- 3.4 Event-Based, Implicit Invocation
- 3.5 Layered Systems
- 3.6 Repositories
- 3.7 Interpreters
- 3.8 Other familiar Architectures
- 3.9 Heterogeneous Architectures

Unit 4: Introduction to Patterns

04 (C03)

- 4.1 Pattern & Design Pattern
- 4.2 What makes a Pattern (GOF)

- 4.3 Describing Design Patterns
- 4.4 Pattern Categories & Relationships between Patterns
- 4.5 Organizing the Catalogue
- 4.6 Patterns and Software Architecture

Unit 5: Study of Design Patterns

14 (CO4)

- 5.1 Creational Patterns-singleton, factory method, abstract factory
- 5.2 Structural Patterns-adapter, decorator, façade
- 5.3 Behavioral Patterns - iterator, observer, strategy, command and state (study of intent, applicability, participants, structure, collaboration, Java Example code, Implementation and consequences)

Unit 6: GRASP (General Responsibility Assignment Software Patterns)

06 (CO5)

- 6.11 Expert, Creator, High Cohesion, Low Coupling
- 6.12 Controller, Polymorphism, Pure Fabrication, Indirection
- 6.13 Don't Talk to Strangers

Unit 7: Study of Frameworks

12 (CO6)

- 7.1 Frameworks as reusable chunks of architecture
- 7.2 The framework lifecycle, development using frameworks
- 7.3 Spring Core Framework
- 7.4 Spring Boot Framework
- 7.5 Microservices with Spring
- 7.6 Web Architectures: Google Web Tool Kit, Spring, Hibernate etc.
- 7.7 Selection of proper framework
- 7.8 Comparing Frameworks
- 7.9 Advantages of Spring
- 7.10 Web based Case Study

Unit 8: Case Study (any one of the web Architecture)

02(CO6)

- 8.1 Take a Framework and find Patterns in the Frame work
- 8.2 Benefits of Patterns in the chosen Framework

Reference Books:

1. Design Patterns – Elements of Reusable Object-oriented Software By E. Gamma, RichardHelm, Ralph Johnson , John Vlissides (GoF)
2. Pattern – Oriented Software Architecture (POSA) Volume 1. By : Frank Buschmann, Regine Meunier, Hans Rohnert, Peter Sommerlad, Michael Stal.
3. Software Architecture in Practice. By Len Bass, Paul Clements, Rick Kazman
4. Applying UML and Patterns By Craig Larman.
5. Head First Design Pattern by Kathy Sierra, Bert Bates, Elisabeth Robson, Eric Freeman Publisher: O'Reilly Media, Inc.
6. Building Microservices-Designing Fine-Grained Systems By Sam Newman Publisher O'Reilly Media
7. Design patterns in Java by Douglas Schmidt Publisher O'Reilly
8. Professional Java Development with the Spring Framework 1st Edition by Rod Johnson, Alef Arendsen, Thomas Risberg, Colin Sampaleanu ; WROX publication

CA-453-MJ: Machine Learning

Lectures: 48 (Credits- 04)

Prerequisites:

- Familiarity with Probability Theory, Multivariable Calculus, Linear Algebra
- Programming in Python (NumPy, SciPy, Pandas, Matplotlib, Seaborn, SciKit-Learn, StatsModel)

Learning Objectives:

1. To understand the fundamental concepts, types, and applications of machine learning.
2. To implement supervised and unsupervised machine learning algorithms using Python.
3. To utilize appropriate machine learning libraries and tools for model development and evaluation.
4. To analyze and apply reinforcement learning techniques for decision-making problems.
5. To understand the fundamental concepts of deep learning and apply neural network architectures to solve complex problems.

Course Outcomes: On completion of this course, students will be able to:

- CO1: Understand key machine learning concepts, including different learning paradigms and their applications.
- CO2: Develop and apply supervised and unsupervised machine learning models to solve real-world problems.
- CO3: Use machine learning libraries to preprocess data, train models, and evaluate their performance.
- CO4: Demonstrate an understanding of reinforcement learning concepts and apply them to decision-making tasks.
- CO5: Develop and implement neural network models for practical applications using deep learning frameworks.

Unit 1: Introduction to Machine Learning

08 (CO1, CO3)

- 1.1 Introduction: Machine Learning, Examples of Machine Learning applications, Training versus Testing, Positive and Negative Class
- 1.2 Types of Learning: Supervised, Unsupervised and Semi-Supervised Learning
- 1.3 Components of Generalization Error (Bias, Variance, underfitting, overfitting)
- 1.4 Metrics for evaluation viz. accuracy, scalability, squared error, likelihood, posterior probability
- 1.5 Creating our own dataset, Importing the dataset, Handling Missing Data, Splitting the dataset into the Training set and Test set
- 1.6 Dimensionality Reduction: Introduction to Dimensionality Reduction, Subset Selection, Feature Scaling and Filtering, Principal Component Analysis

Unit 2: Supervised Machine Learning

14 (CO2, CO3)

- 2.1 Regression - Linear Regression and higher dimensionality, A bi-dimensional example, Univariate Regression, Multivariate Linear Regression, Polynomial Regression, Logistic Regression, Ridge, Lasso and ElasticNet Regression, Robust regression with random sample consensus
- 2.2 Classification - K – Nearest Neighbours (KNN), Naive Bayes Theorem, Support Vector Machine, Decision Tree, Random Tree
- 2.3 Finding the optimal hyper-parameters through grid search
- 2.4 Metrics for Evaluating Classifier Performance - Confusion Matrix, Accuracy, Precision and Recall, F1 Score, Cross-Validation, AUC/ROC Curve

Unit 3: Un-Supervised Machine Learning

12 (CO2, CO3)

- 3.1 Clustering Fundamentals
- 3.2 K-means
- 3.3 Hierarchical Clustering (Agglomerative, Divisive),
- 3.4 Dendrogram

- 3.5 Selecting optimal number of clusters: Within Clusters Sum of Squares (WCSS) by Elbow Method
- 3.6 Association Rules - Support, Confidence and Lift
- 3.7 Apriori Algorithm

Unit 4. Reinforcement Learning

06 (C04)

- 4.1 Introduction- Environment, State, Reward, Policy, Value
- 4.2 Upper Confidence Bound
- 4.3 Thompson Sampling
- 4.4 Q-Learning

Unit 5. Deep Learning

08 (C05)

- 5.1 Introduction to Deep Learning
- 5.2 Artificial Neural Network
- 5.3 Convolution Neural Network
- 5.4 Recurring Neural Network
- 5.5 Generative Adversarial Networks

Reference Books:

- 1. Mitchell, Tom M. "Machine learning. WCB." (1997).
- 2. Rogers, Simon, and Mark Girolami. A first course in machine learning. CRC Press, 2015.
- 3. Friedman, Jerome, Trevor Hastie, and Robert Tibshirani. The elements of statistical learning. Vol.1. Springer, Berlin: Springer series in statistics, 2001.
- 4. Machine learning course material by Andrew Ng, Stanford University
- 5. Sutton, Richard S., and Andrew G. Barto. Reinforcement learning: An introduction. Vol. No. 1. Cambridge: MIT press, 1998.

CA-454-MJP: Lab based on Full Stack Development- I and Machine Learning

No. of Sessions: 12 (Credits-2)

List of Assignments for Full Stack Development – I

Assignment 1

JavaScript

Assignment 2

Basics of Node.js

Assignment 3

Based on NPM concept

Assignment 4

Node.js programs with DB

Assignment 5

Based on Express.js

Assignments of Machine Learning:

1. Write a Python program to Prepare Scatter Plot (Use Forge Dataset / Iris Dataset)
2. Write a Python program to find all null values in a given data set and remove them.
3. Write a Python program the Categorical values in numeric format for a given dataset.
4. Write a Python program to transform data with Principal Component Analysis (PCA)
5. Write a Python program to implement simple Linear Regression for predicting house price.
6. Write a Python program to implement multiple Linear Regression for a given dataset.
7. Write a Python program to implement Polynomial Regression for given dataset.
8. Write a Python program to Implement Naïve Bayes.
9. Write a Python program to Implement Decision Tree whether or not to play tennis.
10. Write a Python program to implement linear SVM.
11. Write a Python program to implement k-nearest neighbors ML algorithm to build prediction model.
12. Write a Python program to implement k-means algorithm on a synthetic dataset.
13. Write a Python program to implement Agglomerative clustering on a synthetic dataset.
14. Create a CNN model and train it on MNIST handwritten digit dataset. Using model, find out the digit written by a hand in a given image.

CA-460-MJ: Advanced Databases

Lectures: 48 (credits: 4)

Prerequisite:

- Knowledge of file system concepts
- Strong foundation of Related database Concepts (Basic & Advanced)

Learning Objectives:

1. Provide an overview of the concept of NoSQL technology.
2. Provide an insight to the different types of NoSQL databases
3. Provide insights on different models of Data Distribution
4. To understand the logic of distributed data processing using MapReduce
5. Make the student capable of making a choice of what database technologies to use, based on their application needs.
6. To get in depth understanding of usage of NoSQL technologies like Document and Graph databases.

Course Outcomes: After completion of this course students will be able to -

CO1: know the NoSQL technology, and understand the difference between RDBMS and NOSQL.

CO2: define, compare and use the four types of NoSQL Databases (Document-oriented, Key-Value Pairs, Column-oriented and Graph).

CO3: evaluate different distribution models, and classify different databased using CAP theorem.

CO4: use the concept of MapReduce for distributed data processing.

CO5: perform hands-on NoSQL database lab assignments such as MongoDB and Neo4J.

Unit 1: Introduction to NoSQL

04 (C01)

- 1.1 Brief History and motivation NoSQL Databases
- 1.2 RDBMS vs NoSQL
- 1.3 Types of NoSQL Database
 - 1.3.1 Key Value Databases
 - 1.3.2 Document Databases
 - 1.3.3 Column Family Databases
 - 1.3.4 Graph Databases
- 1.4 When to use NoSQL

Unit 2: Data Models & Aggregates

04 (C02)

- 2.1 Aggregate concepts
- 2.2 Aggregate Data Model vs Relational Data Model .
- 2.3 Introduction to Aggregates oriented databases
 - 2.3.1 Key Value Database
 - 2.3.2 Document Database
 - 2.3.3 Column Family Database
 - 2.3.4 Graph Database

Unit 3: Distribution Models and CAP Theorem**06 (C03)**

- 3.1 Distributed database basics
- 3.2 Sharding and replication
- 3.3 Master – Slave Replication
- 3.4 Peer to Peer Replication
- 3.5 Combining sharding and replication
- 3.6 CAP Theorem

Unit 4: Map Reduce**04 (C04)**

- 4.1 Basic Map-Reduce
- 4.2 Partitioning and Combining
- 4.3 Composing Map-Reduce Calculations
- 4.4 Simple examples and use cases

Unit 5: Document Database – MongoDB**06 (C05)**

- 5.1 MongoDB Overview
- 5.2 Advantages of MongoDB
- 5.3 Data Modelling in MongoDB
- 5.4 Database and Collection Creation/Update/Deletion
- 5.5 Data Types
- 5.6 Insert / Update / Delete Document
- 5.7 Projection / Limit / Sort Records in MongoDB

Unit 6: Graph Database – Neo4J**06 (C05)**

- 6.1 Overview
- 6.2 Data Model
- 6.3 Building Blocks
- 6.4 Introduction to CQL (Cypher Query Language)
- 6.5 Creating nodes and relationships using CQL
- 6.6 Merge / Set / Delete / Remove / Match / For Each / Where / Count
- 6.7 Return / Order by / Limit / Skip

Reference Books:

1. NoSQL Distilled by Pramod Sadalge, Martin Fowler, Publication: Pearson Education
2. NoSQL for Dummies - A Willy Brand
3. MongoDB Cookbook Second Edition by Cyrus Dasadia, Amol Nayak, PCKT Publication
4. Beginning Neo4j by Chris Kemper, Publication: Apress

CA-461-MJP: Lab based on Advanced Databases

No of Sessions: 12 (Credits-2)

MongoDB Assignments:

1. Assignment on Movie Database
Creating Collections Film and Actor and inserting documents in both the collections.
2. Assignment on Company Database
Creating Collections Employee and Transaction and inserting documents in both the collections
3. Write Queries on Movie Database
4. Write Queries on Company Database

Neo4J Assignments:

1. Create a library database as graph model.
2. Create a song database as graph model
3. Create an employee database as graph model
4. Create a social network database as graph model
5. Write simple queries on all graph models

CA-462-MJ: .NET Programming

Lectures: 30 (Credits-2)

Prerequisites:

- Knowledge of object-oriented programming concepts such as data abstraction, encapsulation, inheritance, and polymorphism.
- Familiarity with programming language such as C++ and/or Java.

Learning Objectives:

1. Learn .NET architecture and its core components.
2. Learn the fundamentals of C# programming.
3. Apply object-oriented programming concepts using C#.
4. Use collections, LINQ, and exception handling effectively.
5. Develop database and web applications using ADO.NET, Entity Framework Core, and ASP.NET Core.

Course Outcomes: On completion of this course, students will be able to:

- C01: Understand features of the .NET platform and C#.
- C02: Develop basic applications using C# and .NET.
- C03: Design object-oriented applications using C#.
- C04: Develop robust applications using collections, LINQ, and exception handling.
- C05: Implement database and web applications using ADO.NET, Entity Framework Core, and ASP.NET Core.

Unit 1: .NET Framework and Architecture

05 (C01)

- 1.1 Introduction to .NET
 - 1.1.1 Overview of .NET Platform (.NET Framework, .NET Core, .NET 6)
 - 1.1.2 .NET Architecture
- 1.2 Core Components
 - 1.2.1. Common Language Runtime (CLR)
 - 1.2.2. Microsoft Intermediate Language (MSIL)
 - 1.2.3. Just-In-Time Compiler (JIT)
- 1.3 .NET Features
 - 1.3.1 Framework Class Library (FCL)
 - 1.3.2 Common Type System (CTS)
 - 1.3.3 Common Language Specification (CLS)
 - 1.3.4 Garbage Collection (GC)

Unit 2: C#.NET Fundamentals

05 (C02)

- 2.1 Basics of C# Language
 - 2.1.1 Structure of Console Application
 - 2.1.2 Namespace
 - 2.1.3 Variables and Data Types
 - 2.1.4 Expressions and Operators

- 2.2 Type Handling
 - 2.2.1 Type Conversion
 - 2.2.2 Boxing and Unboxing
- 2.3 Control and Functions
 - 2.3.1 Flow Control (if, switch, loops)
 - 2.3.2 Functions and Parameters

Unit 3: Object-Oriented Programming with C#

06 (C03)

- 3.1 OOP Concepts
 - 3.1.1 Encapsulation, Inheritance, Polymorphism, Abstraction
- 3.2 Classes and Objects
 - 3.2.1 Class structure and object creation
- 3.3 Class Members
 - 3.3.1 Access Modifiers
 - 3.3.2 Properties
 - 3.3.3 Constructors
- 3.4 Advanced OOP
 - 3.4.1 Interfaces
 - 3.4.2 Method overloading and overriding

Unit 4: Arrays and Exception Handling

06 (C04)

- 4.1 Collections
 - 4.1.1 Arrays (1D, 2D)
 - 4.1.2 List, Dictionary
- 4.2 LINQ Basics
 - 4.2.1 Filtering and Sorting
 - 4.2.2 Simple Queries
- 4.3 Exception Handling
 - 4.3.1 System-defined exceptions
 - 4.3.2 User-defined exceptions
 - 4.3.3 try-catch-finally
- 4.4 File Handling
 - 4.4.1 Reading and writing files
 - 4.4.2 StreamReader and StreamWriter

Unit 5: Database Programming and Web Development

09 (C05)

- 5.1 ADO.NET
 - 5.1.1 Connection and Command
 - 5.1.2 DataReader and DataSet
- 5.2 Entity Framework Core
 - 5.2.1 Introduction to ORM
 - 5.2.2 CRUD Operations
 - 5.2.3 Code First Approach (Basics)
- 5.3 Introduction to ASP.NET Core
 - 5.3.1 Overview and Architecture
 - 5.3.2 MVC Pattern
- 5.4 Web Application Development

- 5.4.1 Controllers and Routing
- 5.4.2 Views and Razor Syntax
- 5.5 Web API Basics
 - 5.5.1 Introduction to Web API
 - 5.5.2 JSON Data Handling

Reference Books:

1. *C# 10 and .NET 6 – Modern Cross-Platform Development* by Mark J. Price, Packt Publication
2. *Pro ASP.NET Core MVC 2* by Adam Freeman, Apress Publication
3. *Programming C#* by Ian Griffiths, O'Reilly Publication
4. *Entity Framework Core in Action* by Jon P Smith, Manning Publication

CA-463-MJP: Lab based on .NET

No of Sessions: 12 (Credits-2)

Assignments:

1. Implementation of C# Fundamentals Using Console Applications
2. Design and Implementation of Object-Oriented Concepts in C#
3. Working with Collections and LINQ in C#
4. Handling Exceptions and File Operations in C#
5. Development of Database Applications Using ADO.NET and Entity Framework Core
6. Development of Web Applications Using ASP.NET Core and Web API

CA-481-OJT: On Job Training

Total duration: 120 Hours (Credits-4)

Learning Objectives:

1. To provide students with practical, hands-on-experience in applying theoretical knowledge to real-world tasks
2. To help students develop and enhance their skills, problem solving abilities and work culture of the industry
3. To foster effective teamwork and collaboration skills
4. To encourage students to build and expand their professional network by interaction with experienced experts and mentors in industry

Course Outcomes: On Completion of this course, student will be able to -

- CO 1: Enhance the knowledge related to various tools and technologies used in industry
- CO 2: Improve the ability to solve complex problems independently and creatively
- CO 3: Effectively utilize critical thinking and analytical skills in tackling real world challenges
- CO 4: Effectively communicate and collaborate skills through interaction with team members and mentors.
- CO 5: Get an experience in working on projects or related working within industry
- CO 6: Develop the ability to document process, design, implementation and testing
- CO 7: Familiar with specific industry domain relevant to internship
- CO 8: Complete projects and tasks as per the predetermined objectives

Guidelines for On Job Training (OJT)

1. Student must start the OJT/Internship immediately after semester-V examination during the vacation.
2. Student are expected to complete the IT related work/project within 120 hours assigned by organization (company/ industry/ consultancy/ institution).
3. The internship work may involve the IT related assignment(s) OR the maintenance of existing project OR the design/development of new project OR equivalent work.
4. College should assign the mentors/guides for students to monitor the progress throughout the OJT.
5. Students have to submit the weekly progress report duly signed by the concern authorities of organization to the assigned mentor.
6. At the end of OJT, students should prepare the documentation and submit a report to the college in prescribed format.
7. After completion, the final presentation and documentation will be evaluated by the examination panel.