



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

(Affiliated to Savitribai Phule Pune University)

Three Years B.C.A. Degree Program in Computer Application

T.Y.B.C.A.

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

NEP 2.0

Course Structure of B.C.A. (Science): 2026-27

T.Y. B.C.A

Year	Semester	Course Type	Course Code	Course Title	Remark	Credit	No. of hours
3	V	Major Core	CA-303-MJ	Operating Systems		2	30
			CA-301-MJ	Web Technology – II		2	30
			CA-307-MJ	Programming in Java I		2	30
			CA-306-MJ	Fundamentals of Data Science		2	30
			CA-304-MJP	Practical based on Web Technology – II		2	4 per batch
			CA-308-MJP	Practical based on Programming in Java I		2	4 per batch
		Major Elective	CA-314-MJE	Blockchain		2	30
			CA-315-MJEP	Practical based on Blockchain		2	4 per batch
			OR				
			CA-318-MJE	Software Testing and Testing Tools		2	30
			CA-319-MJEP	Practical based on Software Testing and Testing Tools		2	4 per batch
		Minor	ELC-341-MN	IoT and Applications		2	30
			OR				
			MTC-341-MN	Probability and Statistics for Data science		2	30
		VSC	CA-322-VSC	Artificial Intelligence		2	30
		FP	CA-331-FP	Project		2	
	VI	Major Core	CA-356-MJ	Software Project Management		2	30
			CA-357-MJ	Mobile Application Development		2	30
			CA-360-MJ	Programming in Java II		2	30
			CA-358-MJ	Data Analytics		2	30
			CA-359-MJP	Practical based on Mobile Application Development		2	4 per batch
			CA-361-MJP	Practical Based on Programming in Java II		2	4 per batch
		Major Elective	CA-364-MJE	UI/UX		2	30
			CA-365-MJEP	Practical Based on UI/UX		2	4 per batch
			OR				
			CS-368-MJE	DevOps with GitHub and Docker		2	30
			CA-369-MJEP	Practical Based on DevOps with GitHub and Docker		2	4 per batch
		VSC	CA-371-VSC	Cloud Computing		2	30
		OJT	CA-381-OJT	On Job Training		4	

CA-303-MJ Operating Systems

Lectures – 30 (Credits – 2)

Prerequisites:

- Basic Computer Architecture concepts.
- Basic algorithms and data structure concepts

Learning Objectives:

1. To understand the concept of operation system and its principle
2. To study the various functions and services provided by operating system
3. To understand the notion of process and threads
4. To learn various process scheduling algorithms
5. To learn various memory management techniques
6. To understand the concept of File system management & disk scheduling

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

C01: learn the structure and architecture of operating systems.

C02: understand services provided by the operating system and how they are provided.

C03: understand the concept of process and thread.

C04: understand and compare various process scheduling algorithms.

C05: understand various memory management techniques and study their advantages and disadvantages.

C06: learn file system management and disk scheduling techniques.

Unit 1: Introduction to Operating Systems

02 (C01, C02)

- 1.1 Operating System Definition
- 1.2 Computer System Architecture – Single processor, Multiprocessor, Clustered
- 1.3 Operating System Structure – Simple, Layered, Microkernels, Modules and Hybrid
- 1.4 Computing Environments- Traditional, mobile, distributed, Client/server, peer to peer computing
- 1.5 Operating System services
- 1.6 User and operating system interface
- 1.7 Booting
- 1.8 System calls
- 1.9 System Programs

Unit 2: Processes and Threads

03 (C03)

- 2.1 Process Concept – The processes, Process states, Process control block
- 2.2 Process Scheduling – Scheduling queues, Schedulers, context switch
- 2.3 Operations on Process – Process creation, Process termination, Synchronization of parent and child
- 2.4 Threads – Overview, Benefits of threads, Multithreading Models

Unit 3: CPU Scheduling

06 (C04, C02)

- 3.1 Basic Concept
 - 3.1.1 CPU-I/O burst

- 3.1.2 CPU scheduler
- 3.1.3 Preemptive scheduling
- 3.1.4 Dispatcher
- 3.1.5 Scheduling Criteria
- 3.2 CPU Scheduling Algorithms
 - 3.2.1 FCFS
 - 3.2.2 SJF
 - 3.2.3 Priority scheduling
 - 3.2.4 Round-robin scheduling
 - 3.2.5 Multiple queue scheduling
 - 3.2.6 Multilevel feedback queue scheduling

Unit 4: Memory Management

12 (C05, C02)

- 4.1 Introduction
 - 4.1.1 Basic hardware
 - 4.1.2 Address binding
 - 4.1.3 Logical versus physical address space
 - 4.1.4 Dynamic loading
 - 4.1.5 Dynamic linking and shared libraries
- 4.2 Swapping
- 4.3 Contiguous Memory Allocation
 - 4.3.1 Memory mapping and protection
 - 4.3.2 Memory allocation
 - 4.3.3 Fragmentation
- 4.4 Segmentation memory management mechanism
 - 4.4.1 Basic concept
 - 4.4.2 Segmentation Hardware
- 4.5 Paging memory management mechanism
 - 4.5.1 Basic Concept
 - 4.5.2 Hardware support
 - 4.5.3 Protection
 - 4.5.4 Shared Pages
- 4.6 Demand paging
 - 4.6.1 Basic concept
 - 4.6.2 Performance of demand paging
- 4.7 Page replacement algorithms
 - 4.7.1 FIFO
 - 4.7.2 Optimal
 - 4.7.3 LRU and MFU
 - 4.7.4 Second chance

Unit 5: Protection and Security

07 (C06, C02)

- 5.1 The security problem
- 5.2 Program threats
 - 5.2.1 Trojan horse
 - 5.2.2 Trap door
 - 5.2.3 Logic bomb
 - 5.2.4 Stack and buffer overflow
 - 5.2.5 Viruses

- 5.3 System and network threats
 - 5.3.1 Worms
 - 5.3.2 Port scanning
 - 5.3.3 Denial of service
- 5.4 Cryptography as a security tool
 - 5.4.1 Encryption
- 5.5 User authentication
- 5.6 Implementing security defenses
- 5.7 Firewalling to protect systems and networks
- 5.8 Computer security classification

Reference Books:

1. Operating System Concepts, Abraham Silberschatz, Peter Galvin, Greg Gagne, Student Edition, Wiley Asia
2. Operating Systems: Internals and Design Principles, William Stallings, Prentice Hall of India.
3. Advanced Concepts in Operating Systems, M Singhal and NG Shivaratri, Tata McGraw Hill Inc, 2001

CA-301-MJ Web Technologies – II

Lectures: 30 (Credits - 2)

Prerequisites:

- Knowledge of HTML, CSS, JavaScript and basics of PHP

Learning Objectives:

1. To understand web techniques for handling forms, server information, and maintaining state using sessions and cookies.
2. To apply PHP for database interaction and manage relational databases using SQL and PEAR DB.
3. To explore XML document structures and integrate PHP with XML using various parsing methods.
4. To implement AJAX for handling asynchronous data exchange and basic database interactions using PHP.
5. To learn the fundamentals of the CodeIgniter PHP framework and write simple programs using its core features.

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

- C01:** Identify and use web techniques such as processing forms, handling server responses, and managing sessions and cookies.
- C02:** Implement database interactions using PHP and perform CRUD operations on relational databases with SQL.
- C03:** Work with XML documents by parsing and manipulating XML data using PHP.
- C04:** Develop AJAX-based scripts to handle asynchronous data exchange and interact with databases using PHP.
- C05:** Implement basic programs using CodeIgniter to understand the MVC structure, session handling, and data management.

Unit 1: Web Techniques

08 (C01)

- 1.1 Variables
- 1.2 Server information
- 1.3 Processing forms
- 1.4 Setting response headers
- 1.5 Maintaining state – session and cookies

Unit 2: Databases

04 (C02)

- 2.1 Using PHP to access a database
- 2.2 Relational databases and SQL
- 2.3 PEAR DB basics

Unit 3: XML and JSON

06 (C03)

- 3.1 Introduction to XML
- 3.2 XML document Structure
- 3.3 PHP and XML
- 3.4 The simple XML extension

- 3.5 Introduction to JSON
- 3.6 JSON Data types
- 3.7 JSON Object
- 3.8 JSON schema
- 3.9 JSON Comparison with XML
- 3.10 Encoding and Decoding JSON in PHP

Unit 4: Ajax

08 (C04)

- 4.1 Introduction of AJAX
- 4.2 AJAX web application model
- 4.3 Performing AJAX validation
- 4.4 Handling XML data using php and AJAX
- 4.5 Handling JSON data using php and AJAX
- 4.6 Connecting database using php and AJAX

Unit 5: PHP framework CodeIgniter

04 (C05)

- 5.1 Basic concept of CodeIgniter
- 5.2 Installing CodeIgniter
- 5.3 Application Architecture
- 5.4 MVC Framework
- 5.5 Libraries Working with databases
- 5.6 Load external JS and CSS page & redirecting from controller
- 5.7 Adding JS and CSS, Page redirection
- 5.8 Loading dynamic data on page & session management, cookies management
- 5.9 Restful API

Reference Books

1. Programming PHP By Rasmus Lerdorf and Kevin Tatroe, O'Reilly publication
2. Beginning PHP5, Wrox publication
3. PHP for Beginners, SPD publication
4. AJAX Black Book Kogent solution
5. Mastering PHP BPB Publication
6. PHP cookbook O'Reilly publication
7. Professional Codeigniter By Thomas Myer, Wrox Publication
8. Codeigniter 2 CookBook By Rob Foster, PACKT Publication

CA-307-MJ Programming in Java I

Lectures: 30 (Credits-2)

Prerequisites: Object Oriented Programming concepts

Learning Objectives:

1. To learn principles of Object-Oriented Programming (OOP).
2. Learn how to write, compile, and run Java programs.
3. To learn implementation of object-oriented concepts.
4. To introduce the principles of inheritance and polymorphism; and demonstrate how they relate to the design of abstract classes.
5. To introduce the concepts of exception handling and file handling.

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

- CO1:** Understand object-oriented concepts.
CO2: Develop and Execute java programs using command line and console input.
CO3: Design and implement object-oriented solutions using classes and objects.
CO4: Apply inheritance, interfaces, and polymorphism in program design.
CO5: Develop robust applications using exception and file handling.

Unit 1: Object oriented programming (OOP) concepts 02 (C01)

- 1.1 Procedure-oriented programming Vs Object-oriented programming
- 1.2 Classes and objects
- 1.3 Abstraction
- 1.4 Inheritance
- 1.5 Polymorphism
- 1.6 Encapsulations

Unit 2: Introduction to Java 06 (C02)

- 2.1 Features of Java
- 2.2 Java Environment – Compiler, Interpreter, JVM
- 2.3 Simple java program
- 2.4 Accepting input using Command line arguments
- 2.5 Accepting input from console (Using BufferedReader and Scanner class)
- 2.6 Declaring single and multi-dimensional arrays

Unit 3: Classes and Objects 08 (C03)

- 3.1 Defining a class
- 3.2 Access Specifiers (public, protected, private, default)
- 3.3 Array of Objects
- 3.4 Constructors, Overloading Constructors and Use of 'this' keyword
- 3.5 static block, static fields and methods
- 3.6 Creating, Accessing and Using Packages
- 3.7 Predefined Classes
 - 3.7.1 Object Class Methods - equals(), toString(), hashCode(), getClass()
 - 3.7.2 String class (Basic methods), StringBuffer class
- 3.8 Wrapper Classes
- 3.9 Concept of Garbage Collection

Unit 4: Inheritance and Interface**08 (C04)**

- 4.1 Inheritance Basics (extends Keyword) and Types of Inheritance
- 4.2 Superclass, Subclass and use of Super Keyword
- 4.3 Method Overriding and runtime polymorphism
- 4.4 Use of abstract class and abstract methods
- 4.5 Use of final keyword related to method and class
- 4.6 Defining and Implementing Interfaces
- 4.7 Runtime polymorphism using interface
- 4.8 Marker and functional interface

Unit 5: Exception and File Handling**06 (C05)**

- 5.1 Exception class, Checked And Unchecked Exception
- 5.2 Catching Exceptions, Multiple catch Blocks
- 5.3 throw, throws and finally keywords
- 5.4 Creating User Defined Exception
- 5.5 Introduction to Files and Streams Input-OutputStream : FileInputStream/OutputStream, DataInput/OutputStream
- 5.6 Reader-Writer : FileReader/Writer, BufferedReader/Writer

Reference Books:

- 1. Core Java Volume I - Fundamentals By Cay S. Horstmann, 11th Edition, Prentice Hall, ISBN 978-0-13-516630-7
- 2. The Complete Reference By Herbert Schildt, 11th Edition, McGraw Hill Education, ISBN 978-260-44023-2
- 3. Java Beginners Guide By Herbert Schildt, 8th Edition, McGraw-Hill Education ISBN 978-1-260-44021-8
- 4. Java 2 programming black books, StevenHorlzner

CA-306-MJ Fundamentals of Data Science

Lectures: 30 (Credits-2)

Prerequisites:

- Problem solving using computers
- Basic mathematics and statistics
- Knowledge of Databases

Course Objectives:

1. To introduce the fundamental concepts, lifecycle, and applications of data science.
2. To develop an understanding of statistical methods for analyzing and interpreting data.
3. To equip students with techniques for data preprocessing, including cleaning, transformation, and reduction.
4. To enable students to explore and visualize data using appropriate tools and techniques for effective decision-making.

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

CO1: Explain core concepts of data science, including data types, sources, and the data science lifecycle.

CO2: Apply descriptive and inferential statistical techniques to analyze datasets and draw meaningful conclusions.

CO3: Perform data preprocessing tasks such as handling missing values, noise reduction, data transformation, and discretization.

CO4: Analyze similarity and dissimilarity measures and identify outliers in datasets.

CO5: Use various data visualization techniques and tools to represent and interpret data effectively.

CO6: Apply exploratory data analysis (EDA) methods to uncover patterns, trends, and insights from data.

Unit 1: Introduction to data science

04 (CO1)

- 1.1 The 3 V's: Volume, Velocity, Variety
- 1.2 Relevance and Applications of Data Science
- 1.3 The Data Science Lifecycle
- 1.4 Types of Data - Structured, semi-structured, Unstructured Data, Problems with unstructured data
- 1.5 Data sources
- 1.6 Open Data, Social Media Data, Multimodal Data, standard datasets
- 1.7 Data Formats - Integers, Floats, Text Data, Text Files, Dense Numerical Arrays, Compressed or Archived Data, CSV Files, JSON Files, XML Files, HTML Files, Tar Files, GZip Files, Zip Files, Image Files: Rasterized, Vectorized, Compressed

Unit 2: Statistical Data Analysis

06 (CO2, CO4)

- 2.1 Role of statistics in data science
- 2.2 Descriptive statistics - Measuring the Frequency, Measuring the Central Tendency: Mean, Median, and Mode, Measuring the Dispersion: Range, Standard deviation, Variance, Interquartile Range
- 2.3 Inferential statistics - Hypothesis testing, Multiple hypothesis testing, Parameter Estimation methods, Measuring Data Similarity and Dissimilarity
- 2.4 Data Matrix versus Dissimilarity Matrix, Proximity Measures for Nominal Attributes,

- Proximity Measures for Binary Attributes, Dissimilarity of Numeric Data: Euclidean, Manhattan, and Minkowski distances, Proximity Measures for Ordinal Attributes
- 2.5 Concept of Outlier, types of outliers, outlier detection methods

Unit 3: Data Preprocessing

10 (C03)

- 3.1 Data Objects and Attribute Types: Attribute - Nominal, Binary, Ordinal, Numeric, Discrete versus Continuous Attributes
- 3.2 Data munging/wrangling operations
- 3.3 Cleaning Data - Missing Values, Noisy Data (Duplicate Entries, Multiple Entries for a Single Entity, Missing Entries, NULLs, Huge Outliers, Out-of- Date Data, Artificial Entries, Irregular Spacings, Formatting Issues - Irregular between Different Tables/Columns, Extra Whitespace, Irregular Capitalization, Inconsistent Delimiters, Irregular NULL Format, Invalid Characters, Incompatible Datetimes)
- 3.4 Data Transformation – Rescaling, Normalizing, Binarizing, Standardizing, Label and One Hot Encoding
- 3.5 Data reduction
- 3.6 Data discretization

Unit 4: Data Visualization

10 (C05, C06)

- 4.1 Introduction to Exploratory Data Analysis
- 4.2 Data visualization and visual encoding
- 4.3 Data visualization libraries
- 4.4 Basic data visualization tools - Histograms, Bar charts/graphs, Scatter plots, Line charts, Area plots, Pie charts, Donut charts
- 4.5 Specialized data visualization tools - Boxplots, Bubble plots, Heat map, Dendrogram, Venn diagram, Treemap, 3D scatter plots

Reference Books:

- 1. Data Science Fundamentals and Practical Approaches, Gypsy Nandi, Rupam Sharma, BPB Publications, 2020.
- 2. The Data Science Handbook, Field Cady, John Wiley & Sons, Inc, 2017
- 3. Data Mining Concepts and Techniques, Third Edition, Jiawei Han, Micheline Kamber, Jian Pei, Morgan Kaufmann, 2012.
- 4. A Hands-On Introduction to Data Science, Chirag Shah, University of Washington Cambridge University Press

CA-304-MJP: Practical Based on Web Technology - II

No. of Sessions: 12 (Credits-2)

Assignments on Web Technology – II

1. Self-Processing , Sticky forms and File upload, COOKIES and SESSIONS
2. PHP Database
3. XML documents, JSON object
4. AJAX
5. PHP framework CodeIgniter

CA-308-MJP Practical based on Programming in Java I

No. of Sessions - 12 (Credits-2)

Assignments on Object Oriented Programming - I

1. Java Tools and IDE, Simple Java Programs
2. Array of Objects and Packages
3. Inheritance and Interfaces
4. Exception and File Handling

CA-314-MJE Blockchain

Lectures: 30 (Credits-2)

Prerequisites:

- Knowledge of Object-Oriented Programming Concepts.
- Knowledge of Python.

Learning Objectives:

1. Learn the fundamentals of blockchain, its architecture, types, and applications.
2. Learn how blockchain functions through hashing, mining, and consensus in a distributed network.
3. Understand the concept, deployment, and significance of smart contracts in decentralized systems.
4. Learn to write, test, and deploy basic smart contracts using the Solidity programming language.

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

- C01:** Understand the basic concepts, structure, and utility of blockchain technology.
- C02:** Describe the internal workings of blockchain including mining, hashing, and consensus protocols
- C03:** Demonstrate the working and real-world application of smart contracts on blockchain platforms.
- C04:** Develop and implement smart contracts using Solidity in the Ethereum environment.

Unit 1: Introduction to Blockchain

08 (C01)

- 1.1 Foundational Computing Concepts (Client-Server systems vs Peer to Peer Systems)
- 1.2 Evolution of Blockchain
- 1.3 Blockchain Vs Database
- 1.4 Essentials of Blockchain (Blockchain generations, types of blockchain, benefits and challenges of blockchain usage)
- 1.5 Types of Networks
- 1.6 Layered Architecture of Blockchain Ecosystem
- 1.7 Components of blockchain
- 1.8 Cryptography (private and public keys, Hashing & Digital Signature)
- 1.9 Consensus Mechanisms
- 1.10 Cryptocurrency, Digital Currency Bitcoin and Ethereum
- 1.11 Smart Contracts
- 1.12 Blockchain use cases

Unit 2: Working of Blockchain

07 (C02)

- 2.1 Understanding SHA256 Hash
- 2.2 Immutable Ledger
- 2.3 Distributed P2P Network
- 2.4 Working of Mining (The NONCE and Cryptographic Puzzle)

- 2.5 Byzantine Fault Tolerance
- 2.6 Consensus Protocols: Proof of Work, Proof of Stake, Défense Against Attackers, Competing Chains
- 2.7 Demonstration of Blockchain (<https://andersbrownworth.com/blockchain>)

Unit 3: Smart Contracts

05 (C03)

- 3.1 Ethereum Network
- 3.2 Smart Contract
- 3.3 Ethereum Virtual Machine, Ether, Gas
- 3.4 DApps
- 3.5 Decentralized Autonomous Organizations (DAO)
- 3.6 Hard and Soft Forks
- 3.7 Initial Coin Offerings
- 3.8 Demo of Smart Contracts

Unit 4: Introduction to Solidity

10 (C04)

- 4.1 Overview of Solidity programming language
- 4.2 Purpose and importance of Solidity in Ethereum smart contract development
- 4.3 Setting Up Development Environment
 - 4.3.1 Tools for Solidity development (e.g., Remix, Truffle, Ganache)
 - 4.3.2 Installing and configuring development tools
 - 4.3.3 Overview of IDEs and their features for Solidity development
- 4.4 Solidity Syntax and Structure
 - 4.4.1 Syntax rules and conventions in Solidity
 - 4.4.2 Structure of a Solidity smart contract
 - 4.4.3 Data types and variables in Solidity
- 4.5 Functions and Modifiers in Solidity
 - 4.5.1 Defining functions in Solidity contracts
 - 4.5.2 Function modifiers and their usage
 - 4.5.3 Understanding visibility specifiers (public, private, external, internal)
- 4.6 Creating a simple smart contract

Reference Books:

- 1. Mastering Blockchain by Imran Bashir, Third Edition, Packt Publication
- 2. Waterhole, The Science of the Blockchain
- 3. Satoshi Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System
- 4. Mastering Ethereum: Building Smart Contracts and DAPPS, by Andreas Antonopoulos, Dr. Gavid Wood, Oreilly Publication

CA-315-MJEP Practical based on Blockchain

No. of Sessions: 12 (Credits-2)

Assignments:

1. Blockchain Working Demonstration
2. Implementation of SHA-256 Hash and menu-driven blockchain program
3. Create and Deploy Simple Smart Contract
4. Implementation of Constructor and Function Modifiers in Solidity

CA-318-MJE Software Testing and Testing Tools

Lectures: 30 (Credits-02)

Prerequisites: None

Learning Objectives:

1. To provide the knowledge of software testing techniques.
2. To understand how testing methods can be used as an effective tool in quality assurance of software.
3. To provide skills to design test case plan for testing software.
4. To provide knowledge of latest testing methods.

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

CO1: To understand fundamental concepts in software testing

CO2: To design test cases and test plans.

CO3: To understand various software testing methods and strategies.

CO4: Understand the different automation tools.

CO5: To understand basic concept of agile testing.

Unit 1: Introduction to Software Testing

04(CO1)

- 1.1 Introduction to Software Testing
- 1.2 Objectives of Software Testing
- 1.3 Basics of Software Testing – faults, errors and failures
- 1.4 Principles of testing
- 1.5 STLC- Software Testing Life Cycle

Unit 2: Test Cases and Test Plan

04(CO2)

- 2.1 Test case Template
- 2.2 Write Test Plan for given application with resources required
- 2.3 Write Test case for login page, registration page and web application using Excel
- 2.4 Prepare Test report for test cases executed

Unit 3: Software Testing Strategies and Techniques

08(CO3)

- 3.1 Functional Testing vs. Non-functional Testing
- 3.2 Types of Testing -Unit Testing, Integration Testing, System Testing, User Acceptance Testing (UAT)
- 3.3 White Box Testing Techniques - Basic path testing, Control Structure Testing
- 3.4 Black Box Testing Techniques – Boundary Value Analysis, Equivalence partitioning
- 3.5 Defect Life Cycle - Classification of Defect, Defect Report

Unit 4: Testing Tool

10(CO4)

- 4.1 How to make use of Automation Tools
- 4.2 Types of Testing Tools
- 4.3 Selenium Framework-Locators, Handling Web Elements, Waits in Selenium, Framework Design, TestNG annotations and reports
- 4.4 Apache JMeter- Performance Testing, Load Testing, Stress Testing

Unit 5: Agile Testing

04(C05)

- 5.1 Introduction to agile testing
- 5.2 Difference between traditional and Agile testing
- 5.3 Agile principles and values
- 5.4 Agile Testing Quadrants

Reference Books:

1. Effective Methods of Software Testing, William E Perry, 3rd Edition, Wiley Publishing Inc.
2. Software Testing Principles and Practices by Srinivasan Desikan, Gopalaswamy Ramesh, Pearson.
3. Software Engineering – A Practitioners Approach, Roger S. Pressman, 7th Edition, Tata McGraw Hill, 20
4. Agile Testing: A Practical Guide for Testers and Agile Teams, Lisa Crispin and Janet Gregory, 1st Edition, Addison-Wesley Professional, 2008

CA-319-MJEP: Practical based on Software Testing and Testing Tools

No. of Sessions - 12 (Credits-2)

Assignments:

1. Assignments on basics of manual testing.
2. Assignments on Automation testing.
3. Assignments on Selenium, JMeter

ELC-341-MN IoT and Applications

Lectures: 30 (Credits-02)

Prerequisites: None

Learning Objectives:

1. Explain the fundamental concepts, architecture, and enabling technologies of the Internet of Things (IoT).
2. Apply IoT design methodology and analyze communication protocols used in IoT systems.
3. Describe the role of cloud platforms in managing and processing IoT data.
4. Design basic IoT applications for domains such as smart homes, agriculture, healthcare, and education.

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

C01: Define the Internet of Things

C02: IoT design methodology and Analyze protocols for communication among IoT devices

C03: Understand cloud platforms for IOT

C04: Design various applications of IOT

Unit 1: Internet of Things Concepts

08(C01)

- 1.1 Definition, Characteristics of IoT
- 1.2 Trends in Adoption of IoT
- 1.3 IoT Devices, IoT Devices Vs Computers
- 1.4 Basic Building Blocks of an IOT Device

Unit 2: Introduction to IOT design methodology and Protocols

10(C02)

- 2.1 Physical Design of IoT: - Things in IoT, Interoperability of IoT Devices,
- 2.2 Sensors and Actuators for IOT
- 2.3 Signal conditioning techniques: bridge amplifier, noise suppression techniques (filters for noise removal and cancellation), Linearization techniques
- 2.4 Need of Analog/Digital Conversion: Block diagrams of successive approximation ADC, Dual Slope ADC, Simultaneous ADC, Comparison of the three based on the parameters. IC examples of the ADC and DAC and their parameters
- 2.5 Logical Design of IoT- IoT functional blocks, IoT Enabling technologies, IoT levels and deployment templates, Applications of IoT
- 2.6 Design Steps
- 2.7 Basics of IoT Networking, Networking Components
- 2.8 IoT Communication Models and IoT Communication APIs

Unit 3: Cloud Platforms for IOT

06(C03)

- 3.1 Sensor nodes and sensor networks
- 3.2 Protocol Standardization for IoT
- 3.3 M2M and WSN Protocols
- 3.4 RFID Protocol,
- 3.5 Modbus Protocol, Zigbee Architecture.
- 3.6 Cloud for IoT
- 3.7 Application of Amazon Web Services for IoT and SkyNet IoT

- 4.1 IOT for Smart home
- 4.2 Smart agriculture using IOT
- 4.3 IOT based health care system
- 4.4 IOT in education
- 4.5 Demonstration of AWS working

Reference Books:

1. Arshdeep Bahga, Vijay Madisetti, "Internet of Things– A hands- on approach", Universities Press, ISBN: 0: 0996025510, 13:978-996025515
2. Olivier Hersent, David Boswarthick, Omar Elloumi, "The Internet of Things: Key Applications and Protocols", 2nd Edition, Wiley Publication, ISBN:978-1-119-99435-0
3. Dawoud Shenouda, Peter Dawoud, "Microcontroller and Smart Home Networks", ISBN: 9788770221566, e-ISBN: 9788770221559
4. Charles Crowell, "Internet of Things for Beginners: An Easy-to-Understand Introduction to IoT", ISBN-13 : 979-8613100194
5. David Hanes, Gonzalo Salgueiro, Robert Barton, Jerome Henry, "IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things", Cisco Press, ISBN-13: 978-1-58714-456-1 ISBN-10: 1-58714-456-5

MTC-341-MN Probability and Statistics for Data Science

Lectures: 30 (Credits-02)

Learning Outcomes

1. Providing the knowledge of basics of Probability and the distribution related
2. Providing understanding of various sampling techniques related various applications in Data Science
3. Introducing the concepts of Hypothesis testing

Course Outcomes:

- CO1.** Students acquired the ability to understand the scenarios to apply suitable probability distribution models
- CO2.** Students have learnt discrete random variables, discrete frequency distributions and the application of Chebyshev's theorem.
- CO3.** Students have learnt continuous random variables and distributions
- CO4.** Students will get knowledge about the sampling distribution
- CO5.** Students have gained sufficient knowledge related to sampling and hypothesis formulation

Unit 1. Basic concepts of Probability

06(C01)

- 1.1 Sample Spaces and Events
- 1.2 Counting
- 1.3 Probability
- 1.4 The Axioms of Probability
- 1.5 Some Elementary Theorems
- 1.6 Conditional Probability
- 1.7 Bayes' Theorem

Unit 2. Discrete random variables, Probability Distributions

08 (C02)

- 2.1 Random Variables
- 2.2 The Binomial Distribution
- 2.3 The Hyper geometric Distribution
- 2.4 The Mean and the Variance of a Probability Distribution
- 2.5 Chebyshev's Theorem
- 2.6 The Poisson Approximation to the Binomial Distribution

Unit 3. Continuous random variable

6 (C03)

- 3.1 Continuous Random Variables
- 3.2 The Normal Distribution
- 3.3 The Normal Approximation to the Binomial Distribution
- 3.4 The Uniform Distribution

Unit 4. Sampling Distribution

6 (C04)

- 4.1 Populations and Samples
- 4.2 The Sampling Distribution of the Mean (known), Central Limit Theorem
- 4.3 The Sampling Distribution of the Mean (unknown), t and Chi-square distribution
- 4.4 The Sampling Distribution of the Variances
- 4.5 Point Estimation
- 4.6 Interval Estimation, Confidence Intervals

Unit 5. Testing of hypotheses**4 (C05)**

5.1 Tests of Hypotheses

5.2 Null Hypotheses and Tests of Hypotheses

5.3 Hypotheses Concerning One Mean

Text Book:T1 - Richard A. Miller & Freund's Probability and Statistics for Engineers, PHI, 8th Ed., 2011**Reference Books :**

1 - Paul L. Meyer, Introductory Probability and Statistics Appl., Second Edition, Addison-Wesley, 1970

2 - Beaver Beaver, Introduction to Probability and Statistics, Thomson, 12th Ed., 2007.

CA-322-VSC Artificial Intelligence

Lectures: 30 (Credits-2)

Prerequisites:

- Basic Programming Concepts

Course Objectives:

1. To learn various types of algorithms useful in Artificial Intelligence (AI).
2. Learn about the various strategies involved in problem-solving.
3. Learn about solving problems with various constraints

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

C01: Identify and apply suitable Intelligent agents for various AI applications

C02: Apply suitable strategies to solve AI problems

C03: Build smart system using different informed search / uninformed search or heuristic approaches

C04: Represent complex problems with expressive language of representation

C05: Understand various AI application like ChatGpt

Unit 1: Introduction to Artificial Intelligence

03 (C01)

- 1.1 Introduction to AI
- 1.2 Comparison of AI, Machine Learning, Deep Learning
- 1.3 Applications of AI
- 1.4 AI Techniques
- 1.5 Intelligent Agents, Agents and Environments, Structure of Agents

Unit 2: Problems, Problem Spaces and search

06 (C02)

- 2.1 Defining problem as a State Space Search
- 2.2 Production System
- 2.3 Problem Characteristics
- 2.4 Search & Control Strategies
- 2.5 Problems – Water Jug problem, Missionary Cannibal Problem, Block words Problem, Monkey & Banana problem

Unit 3: Searching Algorithms

07 (C03)

- 3.1 Uninformed Search Algorithms/Blind Search Techniques
 - 3.1.1 Breadth-first Search
 - 3.1.2 Depth-first Search
- 3.2 Informed (Heuristic) search Techniques
 - 3.2.1 Generate-and-test

- 3.2.2 Simple Hill Climbing
- 3.2.3 Best First Search
- 3.2.4 A* Search, AO* Search
- 3.2.5 Constraint Satisfaction Problem

Unit 4: Knowledge Representation

06 (C04)

- 4.1 Definition of Knowledge
- 4.2 Types of knowledge (Procedural and Declarative knowledge)
- 4.3 Approaches to Knowledge Representation
- 4.4 Knowledge representation using Propositional and Predicate logic
- 4.5 Conversion to clause form
- 4.6 Resolution in Propositional logic

Unit 5: Slot and Filler Structures

05 (C04)

- 5.1 Weak structures
 - 5.1.1 Semantic networks
 - 5.1.2 Frame
- 5.2 Strong structures
 - 5.2.1 Conceptual dependencies
 - 5.2.2 Script

Unit 6: Recent Trends in AI and Applications

03 (C05)

- 6.1 Predictive Analytics (Weather Forecasting)
- 6.2 AI-Powered Chatbots (SBI card chatbot (ILA))
- 6.3 Chat GPT

Reference Books:

1. Artificial Intelligence, Tata McGraw Hill, Elaine Rich and Kevin Knight
2. Artificial Intelligence: A New Synthesis, Nilsson, Elsevier, ISBN 9788181471901
3. Introduction to Artificial Intelligence and Expert System, Dan Patterson, Prentice Hall of India Pvt. Ltd., New Delhi, 1997
4. Artificial Intelligence: A Modern Approach, Russel & Norvig, Pearson Education
5. "Artificial Intelligence: Structures and Strategies for complex problem solving", G. Luger, Fourth Edition, Pearson Education

CA-331-FP Project

No. of sessions – 12 (Credits – 2)

Prerequisites:

- Software Engineering

Learning Objectives:

1. To understand concepts of Project Management.
2. To know how various tools for development and management of software projects are used to carry out various tasks involved.
3. To learn the importance of project documentation.

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

C01: Demonstrate a sound technical knowledge of selected project topic.

C02: Apply techniques for project management.

C03: Create various documents used during the development of the project and a project report.

Project Implementation Guidelines:

1. Students shall choose any topic for project work in consultation with project guide, Project In-charge.
2. The students shall work on a Project in a group of **three** students.
3. Students are expected to work on the chosen project during the entire semester.
4. Students shall undertake application oriented/web-based/database- oriented/research-based work.
5. Students shall successfully implement the chosen work. Only a hypothetical / theoretical study shall not be accepted.
6. Students shall choose any appropriate programming language/ platform, computational techniques and tools in consultation with the guide, In-charge.
7. The faculty members from affiliated college shall act as a project guide for each project group with equal distribution of groups amongst each eligible faculty.
8. The guide shall track and monitor the project progress on a weekly basis by considering the workload of 4 laboratory hours per week.
9. The project work shall be evaluated based on the novelty of the topic, scope of the work, relevance to computer science, adoption of emerging techniques/technologies and its real-world application etc.
10. Students shall prepare a project report with the following contents:
 - a) Title Page
 - b) Certificate
 - c) Index Page detailing description of the following with their sub sections:-
 - Title: A suitable title giving the idea about what work is proposed.
 - Introduction: An introduction to the topic giving proper background of the topic.
 - Requirement Specification: Specify Software/hardware/data requirements.
 - System Design details

: Methodology/Architecture/UML/DFD/Algorithms/protocols
used (whichever is applicable) - System Implementation: Code implementation.

- System Implementation: Code implementation.
- Results: Test Cases/Tables/Figures/Graphs/Screen shots/Reports etc.
- Conclusion and Future Scope: Specify the Final conclusion and future scope.
- References: Books, web links, research articles etc.

11. The Project report should be prepared in a spiral bound form with adequate number of copies.

12. The Project work and report shall be certified by the concerned Project guide and Head of the department.

Recommended Documentation contents:

Sr. No	Index Name	Page No.
I.	Acknowledgement	
II.	College Certificate	
1.	Introduction I. Existing System II. Need for New System	
2.	Problem Definition	
3.	Proposed System: I. Explanation II. Methodology used	
4.	Scope of the System	
5.	Hardware and Software Requirement	
6.	Fact Finding Techniques	
7.	Feasibility Study I. Operational II. Technical III. Economical	
8.	Diagrams: I. E-R Diagram II. UML Diagrams	
9.	Data Dictionary	
10.	Database Designing	
11.	Screen Designing I. I/O Screen Designing	

	II. Output Formats	
12.	Test Cases Design	
13.	Limitations	
14.	Conclusions and Future Enhancements	
15.	Bibliography, References and Published work (Paper/Book Chapter/Copyright/Patent etc.)	

Evaluation Scheme:

Description	Evaluation	Marks
Continuous Evaluation, Progress Report	Internal	25
Presentation & Project Report	External	10
Demonstration of the Project		10
Viva		05
Total		50

Submission of Certified Project Report is mandatory for appearing the Practical Examination (Project).

CA-356-MJ Software Project Management

No. of Hours: 30 (Credits-2)

Course Objectives:

1. To understand the fundamentals of Software Project Management.
2. To introduce Software project planning and management tools.
3. To study software project scheduling and tracking.
4. To know the agile project management.
5. To learn managing people in software project.

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

C01: Comprehend Software Project Management Concepts.

C02: Identify the objectives of activity planning, develop project schedules and differentiate between activity relationships.

C03: Apply Agile Project Management concepts.

C04: Understand and apply software quality metrics and quality improvement techniques.

Unit 1: Project Evaluation and Project Planning

06(C01)

- 1.1 Project and Importance of Project Management
- 1.2 Processes and Knowledge Areas in Project Management (PM)
- 1.3 Activities Covered by Software Project Management
- 1.4 Cost-benefit evaluation and Risk evaluation
- 1.5 Project Planning
- 1.6 Steps for Project Planning

Unit 2: Project Life Cycle and Activity Planning

10(C02)

- 2.1 Project life cycle
- 2.2 PERT and Gantt Charts
- 2.3 Objectives of Activity planning
- 2.4 Project Schedules, Activities, Sequencing and Scheduling
- 2.5 Planning and Formulating Network Model
- 2.6 Activity relationships (FS, SF, SS, FF)
- 2.7 Forward Pass technique
- 2.8 Backward Pass techniques
- 2.9 Critical Path concept and remedies

Unit 3: Agile Project Management

07(C03)

- 3.1 Predictive versus Empirical Management
- 3.2 Comparison between Non-Agile and Agile Project
- 3.3 Three stages of Agile Project
- 3.4 Estimation
- 3.5 Scope Management
- 3.6 Roles and Responsibilities
- 3.7 Scheduling and Tracking

Unit 4: Risk Management and Quality Management

07(CO4)

- 4.1 Risk identification
- 4.2 Risk Assessment
- 4.3 Risk Planning and Management
- 4.4 Defect Prevention
- 4.5 Software Quality Management Standards
- 4.6 Quality Concepts, Software Quality assurances, software reviews, formal technical reviews
- 4.7 Formal approaches to SQA, Statistical Software Quality assurances, Change Management: software Configuration Management

Reference Books:

- 1. Ken Schwaber, "Agile Project Management", Microsoft Press, 2004
- 2. Walker Royce, "Software Project Management", Addison-Wesley, 1998.
- 3. Jalote Pankaj, "Software Project Management in Practice", Addison Wesley Professional, 2002
- 4. Bob Hughes, Mike Cotterell, and Rajib Mall "Software Project Management " software quality management

CA-357-MJ Mobile Application Development

Lectures: 30 (Credits-2)

Prerequisites:

- Basic Knowledge about Java Programming.
- Knowledge about any database management system.

Learning Objectives:

1. Understand the basic features and components of Android and build a simple application.
2. Learn the concept of Activities, their lifecycle, and how to use Intents and Toasts.
3. Explore basic views, layout managers, and menus for designing Android UIs.
4. Understand database handling using SQLite in Android applications.

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

- C01.** Demonstrate the ability to create and run a basic Android app using Android Studio.
- C02.** Develop applications using multiple activities and inter-activity communication with intents.
- C03.** Design user-friendly interfaces using various Views, ViewGroups, and Menus in Android.
- C04.** Implement database operations (insert, update, delete, retrieve) using SQLite in Android apps.

Unit 1: Introduction to Android

05(C01)

- 1.1 Features of Android
- 1.2 Components of Android
- 1.3 Creating your first Android Application

Unit 2: Activities and Intents

07(C02)

- 2.1 Activity and its lifecycle
- 2.2 Intent with its types
- 2.3 Toast, Alert Dialog Box

Unit 3: User Interface- View and Viewgroup

12(C03)

- 3.1 Basic Views: TextView, Button, ImageButton, EditText, CheckBox, ToggleButton, RadioButton, RadioGroup, AutoCompleteTextView, ImageView
- 3.2 List Views: List, Spinner View
- 3.3 View Groups: Linear Layout, Table Layout, Relative Layout, Frame Layout, Constraint Layout, Scroll View
- 3.4 Menus: Options Menu, Context Menu

Unit 4: Databases – SQLite

06(C04)

- 4.1 Introduction to SQLite
- 4.2 SQLite Open Helper and SQLite Database
- 4.3 Creating, opening and closing database
- 4.4 Working with cursors, Insert, Update, Delete

Reference Books:

1. Beginning Android4 Application Development, By Wei-Meng Lee WILEY India Edition WROX Publication
2. Professional Android 4 Application Development, By Reto Meier WROX Publication
3. Head First Android Development: A Brain-Friendly Guide, By David Griffiths and Dawn Griffiths

CA-360-MJ Programming in Java II

Lectures: 30 (Credits-2)

Prerequisites: Knowledge of Core Java

Learning Objectives:

1. To understand different various types of data structures.
2. To develop a game application using multithreading.
3. To learn database programming using Java.
4. To study web development concept using Servlet and webserver.
5. To use Spring Framework and Spring Boot for developing Web application.

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

C01: Implement and analyze different Java Collection framework classes.

C02: Design and develop multithreaded Java applications.

C03: Develop Java applications that connect to relational databases using JDBC.

C04: Understand web applications using Servlets, JSP.

C05: Develop enterprise-level web and REST applications using Spring Framework and Spring Boot.

Unit 1: Collections

05 (C01)

- 1.1 Introduction to the Collection framework
- 1.2 List - ArrayList, LinkedList
- 1.3 Set - HashSet, TreeSet
- 1.4 Map - HashMap and TreeMap
- 1.5 Interfaces such as Iterator, ListIterator, Enumeration
- 1.6 Lambda expression with collection

Unit 2: Multithreading

08 (C02)

- 2.7 Thread concept
- 2.8 Life cycle of thread
- 2.9 Thread priorities
- 2.10 Creating threads - Thread class , Runnable interface
- 2.11 Running multiple threads
- 2.12 Inter thread communication methods – join(), wait(), notify() and notifyAll()
- 2.13 Thread Synchronization

Unit 3: Database Programming

08 (C03)

- 3.1 The design of jdbc
- 3.2 Types of drivers
- 3.3 Statement and PreparedStatement
- 3.4 Executing sql statements, query execution
- 3.5 Metadata – Databasemetadata, ResultSetMetadata
- 3.6 Scrollable and updatable ResultSet

Unit 4: Servlets and JSP**03 (C04)**

- 4.1 Introduction to Servlets
- 4.2 Handling get and post request (HTTP)
- 4.3 Concept of JSP
- 4.4 Understanding architecture of tomcat webserver

Unit 5: Spring and Spring Boot**06 (C05)**

- 5.1 Introduction of Spring framework
- 5.2 Spring Framework Architecture
- 5.3 IoC Container and Dependency Injection (DI)
- 5.4 Annotations
- 5.5 Bean definition
- 5.6 Web MVC framework
- 5.7 MVC form handling
- 5.8 Introduction to Spring Boot
- 5.9 Creating REST Controller
- 5.10 Form Validation

Reference Books:

1. Core Java Volume-I-Fundamentals, Eighth Edition, Cay S. Horstmann, Gary Cornell, Prentice Hall, Sun Microsystems Press
2. Core Java Volume-II-Advanced Features, Eighth Edition, Cay S. Horstmann, Gary Cornell, Prentice Hall, Sun Microsystems Press
3. Getting started with Spring Framework: covers Spring 5 by J Sharma and Ashish Sarin
4. Spring 4 for Developing Enterprise Applications: An End-to-End Approach by Henry H. Liu

CA-358-MJ Data Analytics

Lectures: 30 (Credits-2)

Prerequisites:

- Basic of mathematics and statistics
- Basic programming Knowledge of python
- Knowledge of databases

Learning Objectives:

1. To understand the concept of data analytics, its types, and basic evaluation models used in analyzing data.
2. To explore the modeling process in machine learning and understand different learning approaches and regression techniques.
3. To learn key data mining techniques including frequent pattern mining, association rule generation, and predictive modeling.
4. To analyze social media data using network-based approaches and understand key social media analytics techniques.
5. To understand and apply basic text analytics techniques, including NLP, sentiment analysis, and document summarization.

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

- C01:** Understand Data Analytics and its types, know about different models for Data Analysis
- C02:** Use appropriate models of analysis, assess the quality of input, and derive insight from results.
- C03:** Understand different data mining techniques like classification, prediction, clustering and association rule mining
- C04:** Apply modeling and data analysis techniques to the solution of real-world business problems
- C05:** Analyze data, choose relevant models and algorithms for respective applications

Unit 1: Introduction to Data Analytics

05 (CO 1)

- 1.1 Concept of data analytics
- 1.2 Data analysis vs Data analytics
- 1.3 Types of analytics
 - 1.3.1 Diagnostic Analytics
 - 1.3.2 Predictive Analytics
 - 1.3.3 Prescriptive Analytics
 - 1.3.4 Exploratory Analysis
 - 1.3.5 Mechanistic Analysis
- 1.4 Mathematical models - Concept
- 1.5 Model evaluation
 - 1.5.1 Metrics for evaluating classifiers
 - 1.5.2 Class imbalance - AUC, ROC (Receiver-Operator Characteristic) curves
 - 1.5.3 Evaluating value prediction models

Unit 2: Machine Learning Overview**08 (CO 2)**

- 2.1 Introduction to Machine Learning, deep learning, Artificial intelligence
- 2.2 Applications for machine learning in data science
- 2.3 The modeling processes
 - 2.3.1 Selecting a model
 - 2.3.2 Training the model
 - 2.3.3 Validating the model,
 - 2.3.4 Predicting new observations
- 2.4 Types of machine learning
 - 2.4.1 Supervised learning
 - 2.4.2 Unsupervised learning
 - 2.4.3 Semi-supervised learning
- 2.5 Regression models
 - 2.5.1 Linear Regression
 - 2.5.2 Logistic Regression
 - 2.5.3 Polynomial Regression
- 2.6 Concept of classification, clustering

Unit 3: Mining Frequent Patterns, Associations, and Correlations**08 (CO 3)**

- 3.1 Class/Concept Description: Characterization and Discrimination
- 3.2 Mining Frequent Patterns, Associations, and Correlations
- 3.3 Classification and Regression for Predictive Analysis
- 3.4 Cluster Analysis
- 3.5 Outlier Analysis
- 3.6 Mining frequent patterns - Market Basket Analysis.
- 3.7 Frequent Itemsets, Closed Itemsets, and Association Rules
- 3.8 Frequent Itemset Mining Methods
- 3.9 Apriori Algorithm
- 3.10 Generating Association Rules from Frequent Itemsets
- 3.11 Frequent pattern growth (FP-growth) algorithm

Unit 4: Social Media Analytics**05 (CO 4, CO 5)**

- 4.1 Overview of social media analytics
- 4.2 Social Media Analytics Process, Seven layers of social media analytics, accessing social media data
- 4.3 Key social media analytics methods
- 4.4 Social network analysis
- 4.5 Link prediction, Community detection, Influence maximization, Expert finding, Prediction of trust and distrust among individuals
- 4.6 Challenges to social media analytics

Unit 5: Text Analytics**04 (CO 4, CO 5)**

- 5.1 Introduction to Natural Language Processing
- 5.2 Text Analytic: Tokenization, Bag of words, Word weighting: TF-IDF, n- Grams, stop words, Stemming and lemmatization, synonyms and parts of speech tagging

- 5.3 Sentiment Analysis
- 5.4 Document or text summarization
- 5.5 Trend analytics

Reference Books:

1. Data Science Fundamentals and Practical Approaches, Gypsy Nandi, Rupam Sharma, BPB Publications, 2020.
2. A Hands-On Introduction to Data Science, Chirag Shah, University of Washington Cambridge University Press
3. The Data Science Handbook, Field Cady, John Wiley & Sons, Inc, 2017
4. Data Mining Concepts and Techniques, Jiawei Han, Micheline Kamber, Jian Pei, Morgan Kaufmann, Third Edition, 2012.

CA-359-MJP: Practical based on Mobile Application Development

No. of Sessions: 12 (Credits-2)

Assignments:

1. Introduction to Android
2. Activities and Intents
3. Android User Interface
4. Designing User Interface with Views and Menus
5. SQLite Database

CA-361-MJP Practical based on Programming in Java II

No. of Sessions - 12 (Credits-2)

Assignments on Object Oriented Programming – II:

1. Collection
2. Multithreading
3. Database programming
4. Spring and Spring Boot

Prerequisites:

- Basic Knowledge about Java Programming.
- Knowledge about any database management system.

Learning Objectives:

6. Understand the basic features and components of Android and build a simple application.
7. Learn the concept of Activities, their lifecycle, and how to use Intents and Toasts.
8. Explore basic views, layout managers, and menus for designing Android UIs.
9. Understand database handling using SQLite in Android applications.

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

Unit 1: Foundation of UI/UX Design

05(C01)

- 1.1 Purpose & function
- 1.2 Problem solving
- 1.3 Research & ideation
- 1.4 Prototyping, testing, refinement
- 1.5 UI vs UX
- 1.6 Design principles
- 1.7 User-centered design
- 1.8 Usability & accessibility
- 1.9 Tools & trends
- 1.10 UI Design
- 1.11 Visual Design
- 1.12 Layout & Interface

Unit 2: Information Architecture, Interaction & Content Design

07(C02)

- 2.1 AI design tools
- 2.2 Prompt Crafting
- 2.3 Workflow efficiency
- 2.4 Innovation & future readiness
- 2.5 Structure and Navigation
- 2.6 Content organization
- 2.7 User flows
- 2.8 Information architecture
- 2.9 Usability enhancements

Unit 3: Visual Design, Layout & Design Systems

06(C03)

- 3.1 Grid systems
- 3.2 Alignment & spacing
- 3.3. Visual hierarchy
- 3.4 Responsive design

- 3.5 Legibility & readability
- 3.6 Contrast & hierarchy
- 3.7 Accessibility
- 3.8 Type pairing

Unit 4: Accessibility, Usability & Emerging Technologies

12(C04)

- 4.1 Accessibility
 - 4.1.1 Introduction to accessibility
 - 4.1.2 WCAG principles
 - 4.1.3 Color contrast
 - 4.1.4 Typography accessibility
 - 4.1.5 Keyboard navigation
 - 4.1.6 Screen readers
 - 4.1.7 Accessible forms and buttons
 - 4.1.8 Inclusive design principles
- 4.2 Usability & Evaluation
 - 4.2.1 Usability testing basics
 - 4.2.2 Heuristic evaluation
 - 4.2.3 Usability improvements
- 4.3 AI & Modern UX
 - 4.3.1 Introduction to AI in UX
 - 4.3.2 AI-assisted design workflows
 - 4.3.3 Prompt crafting basics
 - 4.3.4 Design-to-code concepts
- 4.4 Emerging Technologies
 - 4.4.1 Introduction to MCP
 - 4.4.2 AI-driven UI concepts
 - 4.4.3 Role of structured design in AI systems
 - 4.4.4 Future trends in UX and product design
- 4.5 Tools & Industry Trends
 - 4.5.1 Overview of modern UX tools
 - 4.5.2 Role of collaboration tools
 - 4.5.3 Future of UX design

Reference Books:

1. Refactoring UI by Adam Wathan, Steve Schoger
2. Laws of UX: Design Principles for Persuasive and Ethical Products by Jon Yablonski
3. Designing Interfaces by Jennifer Tidwell

CA-365-MJEP Practical Based on UI/UX

No. of Sessions: 12 (Credits-2)

Assignments on UI/UX

Basic UI/UX Understanding

1. Choose any mobile app you use daily and analyze its UI. Identify 5 good design elements and 5 areas for improvement.
2. Create a simple wireframe for a login page of a website using paper or any design tool (Figma, Adobe XD, etc.).
3. Redesign a poorly designed website homepage and explain your improvements.
4. Compare the UI of two similar apps (e.g., food delivery apps) and evaluate which one provides a better user experience.
5. Create a color palette for a travel website and justify your choices.

Wireframing & Prototyping

6. Design low-fidelity wireframes for an e-commerce app (Home, Product Page, Cart).
7. Create a clickable prototype of a student portal using Figma or any prototyping tool.
8. Design a responsive layout for a blog website (mobile + desktop view).
9. Create a user flow diagram for booking a movie ticket online.
10. Design a dashboard UI for a fitness tracking app.

Design Principles & Accessibility

11. Redesign a webpage to improve accessibility (color contrast, font size, navigation).
12. Create UI components (buttons, forms, cards) following accessibility guidelines.
13. Analyze how typography affects readability in any app or website.
14. Design a complete mobile app UI (at least 5 screens) for a problem of your choice (e.g., health, education, travel).
15. Create a case study of your UI/UX project including problem statement, research, wireframes, prototype, and final design.

CA-368-MJE DevOps with GitHub and Docker

Lectures: 30 (Credits - 2)

Prerequisites:

- Basic knowledge of programming concepts, command-line usage and Linux commands

Learning Objectives:

1. To understand the core principles of DevOps and how it connects development and operations teams.
2. To use Git for version control and GitHub for collaborative development.
3. To perform essential Git operations and create simple CI workflows using GitHub Actions.
4. To build, run, and manage Docker containers and images effectively.
5. To create and deploy multi-container applications using Docker Compose.
6. To connect GitHub code to Docker builds and publish images to a registry (Docker Hub).
7. To understand basic DevSecOps principles, particularly secure handling of secrets and credentials in CI/CD pipelines and containerized environments.

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

- C01:** Explain DevOps concepts, utilize Git for version control, manage branches, resolve conflicts, and collaborate via GitHub using pull requests and forks.
- C02:** Design and implement simple CI workflows in GitHub Actions, including triggers, environment variables, and secrets for automated building and testing.
- C03:** Differentiate between VMs and containers, install and use Docker CLI commands, pull/push images from Docker Hub, and write basic Dockerfiles.
- C04:** Build, optimize, and run custom Docker images and multi-container applications using Docker Compose, while handling ports, volumes, and basic debugging.
- C05:** Demonstrate an end-to-end manual DevOps workflow: commit code to GitHub → build Docker image → tag/version → push to Docker Hub → pull and run the containerized application using secure practices (e.g., proper secret handling).

Unit 1: DevOps & Git Fundamentals

06 (C01)

- 1.1 Introduction to DevOps, Overview of DevOps lifecycle and CI/CD pipeline
- 1.2 Introduction to DevSecOps.
- 1.3 Version control concepts: Centralized vs Distributed VCS
- 1.4 Introduction to Git: Installation and basic commands (init, add, commit, status, log)
- 1.5 Branching: branch, checkout, merge, handling simple merge conflicts
- 1.6 Introduction to GitHub: Creating repositories, clone, push, pull
- 1.7 Collaboration basics: Pull Requests, forking, basic team workflows

Unit 2: GitHub Workflows & Basic CI

05 (C02)

- 2.1 GitHub repository structure and good practices (README, documentation)
- 2.2 Introduction to GitHub Actions and Continuous Integration (CI)

2.3 Writing basic workflow YAML files

2.4 Creating a simple CI pipeline (auto-build and test on push) + Security best practices:

Using GitHub Secrets properly (never hardcode passwords/tokens in code or workflows), environment variables, avoiding secret leakage, basic permissions in workflows.

2.5 DevSecOps awareness in CI

Unit 3: Docker Fundamentals

06 (C03)

3.1 Containerization basics: Containers vs Virtual Machines

3.2 Docker architecture and components

3.3 Installing Docker and using basic CLI commands (run, ps, images, stop, rm)

3.4 Working with Docker Hub: Pulling and pushing images

3.5 Writing basic Dockerfiles (FROM, RUN, COPY, CMD/ENTRYPOINT)

3.6 Building and running custom images; introduction to image optimization

Unit 4: Docker Intermediate Concepts

06 (C04)

4.1 Port mapping and exposing container ports

4.2 Docker volumes for persistent data

4.3 Viewing and managing container logs; basic debugging

4.4 Introduction to Docker Compose

4.5 Writing docker-compose.yml files

4.6 Running multi-container applications (e.g: application + database)

Unit 5: Basic GitHub and Docker Integration

07 (C05)

5.1 Overview of a simple DevOps workflow

5.2 Building a Docker image from code stored in GitHub

5.3 Pushing the image to Docker Hub with semantic versioning

5.4 Pulling and running the image from Docker Hub

5.5 Hands-on practice: Complete a basic end-to-end workflow (GitHub → Docker build → Registry → Run)

5.6 Overview of Kubernetes (orchestration, Pods, Deployments, Services) and comparison with Docker Compose. Introduction to Infrastructure as Code (IaC) concepts and benefits.

Reference Books

1. The DevOps Handbook by Gene Kim, Patrick Debois, John Willis, Jez Humble (IT Revolution Press)
2. Docker Deep Dive by Nigel Poulton
3. GitHub Actions Cookbook (or [official GitHub documentation](#))
4. Official Docker Documentation (docs.docker.com)
5. Official GitHub Actions Documentation (docs.github.com/en/actions)

CA-369-MJEP Practical Based on DevOps with GitHub and Docker

No. of Sessions: 12 (Credits-2)

Assignments on DevOps with GitHub and Docker:

1. Git & GitHub Fundamentals Install Git, create a local repo, perform init/add/commit/log, create & merge branches, and resolve simple merge conflicts.
2. Fork & Pull Request Workflow Fork a public repo, clone it locally, make changes, push them, and create a Pull Request.
3. Basic Dockerfile Write a Dockerfile for a static site or Python app, build the image, and run the container with port mapping.
4. Multi-Stage Docker Build Create a multi-stage Dockerfile, build an optimized image, and compare the size before and after optimization.
5. Container Management Practice port mapping, environment variables, volume creation & mounting, and viewing container logs.
6. Docker Compose Write a docker-compose.yml file, set up a Python app + MySQL service, and run the multi-container application.
7. Push Image to Registry Tag a Docker image, push it to Docker Hub, and pull & run it on another system.
8. GitHub Actions Create a simple GitHub Actions workflow that triggers on push and automatically builds a Docker image.
9. GitHub Actions – Continuous Deployment (CD) Extend the GitHub Actions workflow so that on every push to the main branch, it automatically builds the Docker image, pushes it to Docker Hub, and deploys the updated application to a development server using SSH and docker compose up -d. Use GitHub Secrets for Docker credentials and SSH keys. Demonstrate how a code change triggers automatic deployment with minimal manual intervention.
10. **Mini Project:**
Containerize a previous semester project, push the code to GitHub, build & push the Docker image to a registry, and document the complete setup in the README.

CA-371-VSC Cloud Computing

Lectures: 30 (Credits-2)

Prerequisites:

- Computer Network

Learning Objectives:

6. To understand the concept and evolution of cloud computing.
7. To learn virtualization technology and its role in cloud computing.
8. To explore storage architecture and how to analyze and plan storage solutions.
9. To identify major cloud service providers and their offerings.
10. To understand the design of security architecture in cloud computing.

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

- CO1: Explain the fundamental concepts of cloud computing, its models, features, and infrastructure components.
- CO2: Explore virtualization technologies, their implementation.
- CO3: Design and assess storage networks for cloud computing, considering virtualization and storage technologies.
- CO4: Identify, evaluate, and deploy cloud services and applications provided by leading cloud platforms.
- CO5: Explain cloud security concerns, risk management practices, and security architecture principles in a cloud environment.

Unit 1: Introduction to Cloud Computing

05 (CO1)

- 1.1 Cloud computing Overview
- 1.2 Types of Cloud – public, private, hybrid, community
- 1.3 Layers of Cloud – IaaS, PaaS, SaaS
- 1.4 Features of a Cloud Computing
- 1.5 Cloud Infrastructure Components

Unit 2: Abstraction and Virtualization

06 (CO2)

- 2.1 Virtualization Technology
- 2.2 Types of Virtualization
- 2.3 Benefits of Virtualization
- 2.4 Virtualization for Data Center Automation
- 2.5 Hypervisors and Types of Hypervisors
- 2.6 Virtual Machine – System and Process VM
- 2.7 Virtual Machines provision and manageability
- 2.8 Load Balancing
- 2.9 Creating Virtual Machine (Demo)

Unit 3: Storage Network Design

07 (CO4)

- 3.1 Architecture of storage, analysis and planning
- 3.2 Storage network design considerations - NAS and FC SANs, hybrid storage networking technologies (iSCSI, FCIP, FCoE)

3.3 Design for storage virtualization in cloud computing

3.4 Host system design considerations

Unit 4: Deploying Applications and cloud services

06 (C05)

4.1 Cloud service providers

4.2 Microsoft Cloud Services

4.3 Google Cloud Applications

4.4 Exploring SaaS – salesforce.com

4.5 Accessing Storage space on cloud (Demo)

Unit 5: Security in the Cloud

06 (C05)

5.1 Security Architecture Design

5.2 Cloud Security Challenges and Risks

5.3 Software-as-a-Service Security

5.4 Risk Management – Security Monitoring

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-381-OJT On Job Training

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. Students who have appeared for the 4th semester may commence their OJT during the summer vacation. However, they are permitted to complete only up to 80% of the required OJT hours during this period. The remaining 20% must be completed during the winter vacation or in the 6th semester with the same organization.
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