**AGGARWAL COLLEGE BALLABGARH**

**Class-M.C.A.-Ist sem (Lab 1)**

**Subject- Java Programming, Computer Graphics**

**JAVA ASSIGNMENT**

**Week-1**

1. Write a simple java program to print “Hello” word on the screen.
2. Write a simple command line argument program.
3. W.A.P swapping of two number.

**Week-2**

1. W.A.P to compute the area of circle.
2. W.A.P to find the perimeter & square.
3. W.A.P to generate this output

\*

\* \*

\* \* \*

\* \* \* \*

\* \* \* \* \*

**Week-3**

7. W.A.P to generate this output

0

1 2

3 4 5

1. 7 8 9
2. W.A.P to print this output

1

2 2

3 3 3

4 4 4 4

1. W.A.P to calculate the sum of digit of a given number.

**Week-4**

1. W.A.P to calculate the Fibonacci series using do while.
2. W.A.P to display volume of a Box.
3. W.A. Simple Constructor program.

**Week-5**

1. W.A.P of parameterized constructor.
2. W.A.P of Method overriding
3. W.A.P of Method overloading.

**Week-6**

1. W.A.P to show use of Abstract class.
2. W.A.P of single inheritance.
3. W.A.P of multiple inheritance.

**Week-7**

1. W.A.P of hierarchical inheritance.
2. W.A.P to create a single interface.
3. W.A.P to create your own package.

**Week-8**

1. W.A.P of Exception Handling.
2. W.A.P to show multiple catch block.
3. W.A.P to show multiple try block.

**Week-9**

1. W.A.P to show use of exception handling with finally method.
2. W.A.P to show use of a string
3. W.A.P to show use of a string buffer.

**Week-10**

1. W.A.P to show use of a vector class.
2. W.A.P to show multi threading.
3. W.A.P to show use of simple applet.

**Week-11**

1. W.A.P to draw a hut using applet with graphics.
2. W.A.P to draw a human face using applet with graphics.
3. W.A.P to draw a light house using applet with graphics.

**Week-12**

1. W.A.P to read a file.
2. W.A.P to write a file.
3. W.A.P to reverse of already exist file.
4. W.A.P to accept a no. from used and count the no. of digits.
5. W.A.P check no. is Armstrong or not.

**AGGARWAL COLLEGE BALLABGARH**

**Class-M.C.A.-Ist sem (Lab 2)**

**Digital Design and Computer Architecture, Advance Data Structure using C++/Java**

**WEEK -1**

## **What is a Number System?**

*A Number system or* ***numeral system*** *is defined as an elementary system to express numbers and figures. It is the unique way of representing of numbers in arithmetic and algebraic structure.*

## **Types of Number Systems**

Based on the base value and the number of allowed digits, number systems are of many types. The four common types of Number systems are:

* **Decimal Number System**
* **Binary Number System**
* **Octal Number System**
* **Hexadecimal Number System**

## **Decimal Number System**

A number system with a base value of 10 is termed a Decimal number system. It uses 10 digits i.e. 0-9 for the creation of numbers. Here, each digit in the number is at a specific place with a place value of a product of different powers of 10. Here, the place value is termed from right to left as the first place value called units, second to the left as Tens, so on Hundreds, Thousands, etc. Here, units have a place value of 100, tens have a place value of 101, hundreds as 102, thousands as 103, and so on.

***For example, 12265 has place values as,***

*(1 × 104) + (2 × 103) + (2 × 102) + (6 × 101) + (5 × 100)*

*= (1 × 10000) + (2 × 1000) + (2 × 100) + (6 × 10) + (5 × 1)*

*= 10000 + 2000 + 200 + 60 + 5*

*= 12265*

**WEEK-2**

## **Binary Number System**

A number System with a base value of 2 is termed a Binary number system. It uses 2 digits i.e. 0 and 1 for the creation of numbers. The numbers formed using these two digits are termed Binary Numbers. The binary number system is very useful in electronic devices and computer systems because it can be easily performed using just two states ON and OFF i.e. 0 and 1.

Decimal Numbers 0-9 are represented in binary as 0, 1, 10, 11, 100, 101, 110, 111, 1000, and 1001

For example, 14 can be written as 1110, 19 can be written as 10011, and 50 can be written as 110010.

## **Octal Number System**

Octal Number System is one in which the base value is 8. It uses 8 digits i.e. 0-7 for the creation of Octal Numbers. Octal Numbers can be converted to Decimal values by multiplying each digit with the place value and then adding the result. Here the place values are 80, 81, and 82. Octal Numbers are useful for the representation of UTF8 Numbers. Example,

*(81)10 can be written as (121)8*

*(125)10 can be written as (175)8*

## **Hexadecimal Number System**

A number System with a base value of 16 is known as Hexadecimal Number System. It uses 16 digits for the creation of its numbers. Digits from 0-9 are taken like the digits in the decimal number system but the digits from 10-15 are represented as A-F i.e. 10 is represented as A, 11 as B, 12 as C, 13 as D, 14 as E, and 15 as F. Hexadecimal Numbers are useful for handling memory address locations. Examples,

*(185)10  can be written as (B9)16*

*(5440)10  can be written as (1540)16*

*(4265)10  can be written as (10A9)16*

**WEEK-3**

* WAP to convert Decimal to Binary number.
* WAP to convert Binary to Decimal number
* WAP to convert Decimal to Octal number
* WAP to convert Octal to Decimal number

**WEEK-4**

* WAP to convert Decimal to Hexadecimal number.
* WAP to convert Hexadecimal to Decimal number
* WAP to convert Binary to Grey number
* WAP to convert Grey to Binary number
* .

**WEEK-5**

* WAP to implement AND Gate.
* WAP to implement OR Gate.
* WAP to implement NOT Gate.
* WAP to implement XOR Gate.
* WAP to implement EX-NOR Gate.

**WEEK-6**

* WAP to implement Half Adder.
* WAP to implement Full Adder.
* WAP to implement Half comparator.
* WAP to implement Full comparator.
* WAP to implement Multiplexer.

**WEEK-7**

* WAP in java to implement Array.
* WAP in java to implement Array with user inputs.
* WAP in java to implement addition and multiplication of two arrays.
* WAP in java to implement Stack using array.
* WAP in java to implement Stack for using Push and Pop operations.

**WEEK-8**

* WAP in C++ to create Strings.
* WAP in C++ to use String concatenation function.
* WAP in C++ to use String reverse and copy function.
* WAP in C++ to compare functions in Strings.

**WEEK-9**

* Write Java Program to Implement Binary Tree.
* Write Java Program to Implement Binary Search Tree.
* Write Java Program to Implement AVL Tree.
* Write Java Program to Implement AVL Tree with left rotation for Insertion.
* Write Java Program to Implement AVL Tree with right rotation for Insertion.

**WEEK-10**

* Write Java Program to Implement AVL Tree with left right rotation for Insertion.
* Write Java Program to Implement AVL Tree with right left rotation for Insertion.
* Write Java Program to Implement insertion operation in Binary Search Tree.
* Write Java Program to to search an element in Binary Search Tree.
* WAP in java to represent a graph using Linked List.

**WEEK-11**

* WAP in java to represent a graph using Adjacency Matrix.
* WAP in java to represent a graph using Incidence List.
* WAP in java to represent a graph using 2D Array.
* WAP in java to implement Adjacency List.
* WAP to use Except and Between operators.

**WEEK-12**

* WAP in C++ to implement Travelling Salesman Problem using Dynamic Programming.
* WAP in C++ to implement Travelling Salesman Problem using Backtracking.
* WAP in C++ to find shortest path using Floyd Warshall Algorithm.
* WAP in java to implement Naive String Operations.
* WAP in java to implement Advance String operations.

**AGGARWAL COLLEGE BALLABGARH**

**Class-M.C.A.-2nd sem (Lab 1)**

**Advance Object Technology, Theory of Computation & Software Engineering**

**WEEK -1**

* WAP in JavaScript to print Hello World.
* WAP in JavaScript to check whether it is odd or even.
* WAP in JavaScript to find out square root of a given number.
* WAP in JavaScript to show difference between alert(), prompt() and document.write().
* WAP in JavaScript to display multiplication table of a given number.
* WAP in JavaScript to make simple calculator.

**WEEK-2**

* WAP in JavaScript to find the factorial of a given number.
* WAP in JavaScript to implement external javascript file with html.
* WAP in JavaScript to swap 2 variables.
* WAP in JavaScript to check whether given number is positive, negative or zero.
* WAP in JavaScript to check whether a number is prime or not.
* WAP in JavaScript to create array of 7 numbers.

**WEEK-3**

* WAP in java that works as a simple calculator. Use a grid layout to arrange buttons for the digits and for the +, -,\*, % operations.
* WAP in java to show demo of border layout with use of buttons.
* Create a simple Swing application with a JFrame, JLabel, and JButton.
* Write a program to display a message dialog box using JOptionPane.
* Create a Swing application with a JTextField and a JButton to display the entered text.

**WEEK-4**

* Create a Swing application using the FlowLayout manager to arrange components.
* Write a program using the BorderLayout manager to arrange components.
* Create a program to generate an XML file from a database.
* Create a program to check if an XML file is well-formed.

**WEEK-5**

* Introduction to Finite Automata
* Design a Deterministic Finite Automaton (DFA) to recognize the language of all strings ending with "01".
* Construct a Non-Deterministic Finite Automaton (NFA) to recognize the language of all strings containing the substring "110".
* Convert an NFA to a DFA for the language of all strings containing only 0s and 1s.
* Design a DFA to recognize the language of all strings with an even number of 1s.

**WEEK-6**

* Construct an NFA to recognize the language of all strings with a length that is a multiple of 3.
* Regular Expressions and Language
* Write a regular expression to describe the language of all strings containing only 0s and 1s, with an even number of 1s.
* Design a regular expression to recognize the language of all strings ending with "ab".

**WEEK-7**

* Construct a regular expression to describe the language of all strings containing the substring "aba".
* Write a regular expression to describe the language of all strings with a length of at least 2.
* Design a regular expression to recognize the language of all strings containing only lowercase letters.
* Introduction to Context-Free Grammars

**WEEK-8**

* Write a context-free grammar to generate the language of all strings of the form "a^n b^n", where n ≥ 0.
* Design a context-free grammar to recognize the language of all strings containing only 0s and 1s, with an equal number of 0s and 1s.
* Construct a context-free grammar to generate the language of all strings of the form "a^n b^m", where n, m ≥ 0.
* Write a context-free grammar to describe the language of all strings containing only lowercase letters, with a length of at least 2.

**WEEK-9**

* Design a context-free grammar to recognize the language of all strings containing only 0s and 1s, with a length that is a multiple of 3.WAP in java to implement Array with user inputs.
* Introduction to Pushdown Automata
* Design a Pushdown Automaton (PDA) to recognize the language of all strings of the form "a^n b^n", where n ≥ 0.
* Construct a PDA to recognize the language of all strings containing only 0s and 1s, with an equal number of 0s and 1s.

**WEEK-10**

* Write a PDA to describe the language of all strings containing only lowercase letters, with a length of at least 2.
* Design a PDA to recognize the language of all strings containing only 0s and 1s, with a length that is a multiple of 3.
* Construct a PDA to generate the language of all strings of the form "a^n b^m", where n, m ≥ 0.

**WEEK-11**

* Design a simple banking system with multiple branches and ATMs.
* Create a high-level architecture for an e-commerce website.
* Design a database schema for a social media platform.
* Create a unit test for a simple calculator program.
* Write a program to find the maximum and minimum values in an array.

**WEEK-12**

* Describe the Agile development methodology and its benefits.
* Explain the Scrum framework and its roles (Product Owner, Scrum Master, Development Team).
* Create a sample project plan using Gantt charts or a similar tool.
* Describe the concept of velocity in Agile and how it's used.
* Explain the importance of continuous integration and continuous deployment (CI/CD) in software development.

**AGGARWAL COLLEGE BALLABGARH**

**Class-M.C.A.-2nd Sem (Lab2)**

**Operating System and Shell Programming, Advance Database System and Data Mining.**

**WEEK -1**

* Use of basic Unix Shell Commands: ls, mkdir, rmdir, cd, cat, banner, touch, file, wc, sort, cut, grep, dd, dfspace, du, ulimit.
* Commands related to inode, I/O redirection, piping, process control commands, mails.
* Write a shell script to display "Hello, World!" on the screen.
* Write a script that takes two numbers as input and performs addition, subtraction, multiplication, and division.
* Write a script that asks for the user's name and greets them.

**WEEK-2**

1.Shell Programming: shell script exercise based on following:  Interactive shell script 

* Positional parameters 
* Arithmetic
* If-then-fi, if-then-else-fi, nested if-else 
* Logical operators 
* Else + if equals elif,
* case structure 
* While ,for loop 
* Meta characters

**WEEK-3**

* Write a script to replace a word from sentence
* Write a script to find smallest and largest element in array
* Write a script to check whether a given number is even or odd.
* Write a script that checks if a file exists in the current directory and prints an appropriate message.
* Write a script to calculate the factorial of a given number.

**WEEK-4**

* Write a script that checks if a number is prime.
* Write a script to reverse a given string.
* Write a script that logs system uptime and saves it to a file every hour using cron.
* Would you like detailed solutions for any of these?

**WEEK-5**

* Write a script to count the number of words in a given text file.
* Write a script to find the largest of three numbers entered by the user.
* Write a script that monitors a specific process and alerts if it is not running.
* Write a script that renames all .txt files in a directory by adding a timestamp
* Write a script for running a command at regular interval

s

**WEEK-6**

* Write a script to replace pattern in file.
* Write a script to change permission in file
* Write a script to display the top 10 CPU consuming Processes
* Write a script to count number of file in system
* Write a script for automated backup with bash

**WEEK-7**

* To write and optimize complex SQL queries for data retrieval.
* Create a scenario with deadlock and resolve it.
* To explore indexing techniques and optimize queries for large datasets.
* Optimize queries with and without indexes and analyze execution plans.
* Experiment with indexing strategies and discuss the trade-offs.

**WEEK-8**

* To write and optimize complex SQL queries for data retrieval.
* Implement a star schema or snowflake schema for a data warehouse.
* Perform OLAP operations like roll-up, drill-down, slicing, and dicing.
* Use SQL to perform analytical queries on the data warehouse.

**WEEK-9**

* Set up a distributed database with multiple nodes.
* Implement data replication and partitioning strategies.
* Perform queries in a distributed setting and observe the impact on performance.
* Create collections and insert documents in MongoDB

**WEEK-10**

* To implement security mechanisms for a database system.
* Implement user authentication and authorization.
* Implement audit logging for sensitive operations.
* Set up regular backups using different types.

**WEEK-11**

* To implement a backup and recovery strategy in a database system.
* Set up regular backups using different types.
* Perform a database restore operation and test consistency.
* Implement point-in-time recovery and test the backup process.

**WEEK-12**

* Set up an RDS instance in AWS or Google Cloud.
* Create nodes and relationships in a graph database.
* Perform graph traversal queries using Cypher.
* Analyze social network data, recommendation systems, or transportation networks.

**AGGARWAL COLLEGE BALLABGARH**

**Class-M.C.A.-3rd sem (Lab 1)**

**Android Mobile Application Development, Data Mining & Big Data Analytics**

**WEEK -1**

**Android** is a complete set of software for mobile devices such as tablet computers, notebooks, smartphones, electronic book readers, set-top boxes etc. It contains a **linux-based Operating System**, **middleware** and **key mobile applications**.

It can be thought of as a mobile operating system. But it is not limited to mobile only. It is currently used in various devices such as mobiles, tablets, televisions etc. The goal of android project is to create a successful real-world product that improves the mobile experience for end users.

## Create the app project

1. Open Android Studio if it is not already opened.
2. In the main Welcome to Android Studio window, click "Start a new Android Studio project".
3. You will be asked to select a project Template and you should select "Empty Activity" and click the Next button.
4. In the "Configure your Project" window:

* Enter Hello World for the Application name,
* Leave the package name as it is,
* Choose a location to save the project,
* Ensure that Java is selected as the language,
* The Minimum SDK can be API 18. If you want to run your app on a physical device that is running an older version of Android, then change this setting to be compatible with your device.

**WEEK-2**

## Create the app project

## Explore the Project > Android pane

## Explore the Gradle Scripts folder

## Explore the app and res folders

## Explore the manifests folder

**WEEK-3**

* WAP to create Hello World program
* WAP to change the background of an activity
* WAP to create multiple activity within a program
* WAP to change the image on a screen

**WEEK-4**

* WAP to create alert dialog box in Android.
* WAP to create button and radio group in android.
* WAP to display toast message.
* WAP to implement fragment views.

**WEEK-5**

* WAP to create listview in android.
* WAP to create activity using fragment.
* WAP to create basic calculator.
* WAP to use Adapter.
* WAP to use Advance Adapter.

**WEEK-6**

* WAP to use create command.
* WAP to use insert command.
* WAP to use commit command.
* WAP to use select command.
* WAP to use tab command.

**WEEK-7**

* WAP to use where clause command.
* WAP to use rename command.
* WAP to use description command.
* WAP to use order by command
* WAP to use update command

**WEEK-8**

* WAP to use alter command.
* WAP to use drop command.
* WAP to use double table command.
* WAP to use multiply and divide operator.
* WAP to use sysdate.

**WEEK-9**

* WAP to use multiple group function command.
* WAP to create multiple tables using create command.
* WAP to perform different mathematical functions on above created tables.
* WAP to perform Join operations on two tables.
* WAP to perform Left Join and Right Join operations on two tables.

**WEEK-10**

* WAP to perform Full Join and Outer Join operations on two tables.
* WAP to perform Date functions.
* WAP to perform String functions.
* WAP to perform LTrim functions.
* WAP to perform RTrim functions.

**WEEK-11**

* WAP to use In, Like and Not operators.
* WAP to use And, Or and Not Equal Operator.
* WAP to use Union and Union All operators.
* WAP to use Except and Between operators.

**AGGARWAL COLLEGE BALLABGARH**

**Class-M.C.A.-3rd Sem (Lab 2)**

**Software Testing,Artificial Intelligence, Network Programming**

**Week-1:**

 **Write test cases** for a simple login module (valid and invalid scenarios).

 **Perform boundary value analysis** for an input field accepting numbers between 1 and 100.

 **Implement equivalence partitioning** on a registration form with different input conditions.

 **Test a simple calculator application** using decision table testing.

**Week-2:**

 **Perform statement, branch, and path coverage** on a simple C/PHP function.

 **Use JUnit (or PHPUnit) to test a simple function** in Java/PHP.

 **Create a test plan document** for an e-commerce website.

 **Write bug reports** based on test execution.

**Week-3:**

 **Automate a login test case** using Selenium WebDriver with Java/Python.

 **Extract data from a web page** using Selenium and validate content.

 **Perform a basic load test using JMeter** on a sample web page.

 **Analyze JMeter reports** and identify performance bottlenecks.

**Week-4:**

 **Write SQL queries to validate data integrity** after an application operation.

 **Perform backend testing** by checking database constraints and triggers.

 **Perform SQL Injection testing** on a login page.

 **Test for XSS vulnerabilities** on an input form.

**Week-5:**

 I**mplement a simple chatbot using rule-based logic** in Python.

 **Implement Breadth-First Search (BFS) and Depth-First Search (DFS)** on a graph.

 *Implement the A Search Algorithm*\* to solve the 8-puzzle problem.

 **Solve the Traveling Salesman Problem (TSP) using Genetic Algorithm (GA).**

**Week-6:**

 **Implement Propositional Logic using Python (Truth Tables, AND, OR, NOT).**

 **Solve a Sudoku puzzle using Constraint Satisfaction Problem (CSP) techniques.**

 **Implement a Linear Regression model** using scikit-learn.

 **Implement a Decision Tree Classifier** for a dataset (Iris, Titanic, etc.).

**Week-7:**

 **Implement K-Means Clustering** for customer segmentation.

 **Use PCA (Principal Component Analysis) for dimensionality reduction.**

 **Perform Tokenization, Stemming, and Lemmatization** on a text dataset

.

 **Build a simple sentiment analysis model** using Naïve Bayes.

**Week-8:**

 **Implement Q-Learning Algorithm** for solving the Grid World problem.

 **Train an AI agent to play a simple game** using OpenAI Gym.

 **Build a Chatbot** using NLP techniques.

 **Develop a Recommendation System** using Collaborative Filtering.

**Week-9:**

 **Write a simple TCP client-server program** in Python/C using sockets.

 **Write a simple UDP client-server program** and compare it with TCP.

 **Implement a multi-threaded TCP server** that handles multiple clients concurrently.

 **Implement a simple chat application** using sockets.

**Week-10:**

 **Implement a file transfer system** using TCP sockets.

 **Develop a remote command execution program** using socket programming.

 **Write a simple HTTP server** in Python using Flask or in Java using HTTPServer.

 **Implement a WebSocket-based real-time chat application.**

**Week-11:**

 **Secure TCP communication using OpenSSL or Python’s SSL module.**

 **Create a basic HTTPS client to fetch data from a secure website.**

 **Capture and analyze network packets** using Scapy/Wireshark.

 **Write a simple packet sniffer** in Python using socket and struct.

**Week-12:**

 **Develop a basic peer-to-peer file-sharing system.**

 **Implement a simple decentralized messaging system.**

 **Measure network latency and bandwidth using Python/Java.**

 **Implement a simple load balancer** to distribute requests among multiple servers.

**AGGARWAL COLLEGE BALLABGARH**

**Class-M.C.A.-4th sem (Lab 1)**

**Advance Software Engineering, IoT & Sensor, Web Development Using PHP**

**Advance Software Engineering**

**Week 1: Introduction to Software Development**

1. WAP to Identify and Define Software Development Life Cycle (SDLC) Phases for a Given Project
2. WAP to Create a Software Requirement Specification (SRS) Document for an Online Shopping System

**Week 2: UML Diagrams & System Design**

1. WAP to Draw a Use Case Diagram for a Library Management System
2. WAP to Draw a Class Diagram for a Banking System

**Week 3: Data Flow and Entity-Relationship Modeling**

1. WAP to Create a Data Flow Diagram (DFD) for a Hospital Management System
2. WAP to Design an Entity-Relationship (ER) Model for an E-Commerce Application

**Week 4: Software Process Models**

1. WAP to Explain the Waterfall and Agile Model with a Case Study
2. WAP to Compare Spiral and Prototype Models with Examples

**Week 5: Coding Standards & Documentation**

1. WAP to Implement and Document a Simple Calculator using C/Python with Coding Standards
2. WAP to Generate a Technical Report for an Inventory Management System

**Week 6: Software Testing Techniques**

1. WAP to Perform Unit Testing on a Simple Java/Python Program
2. WAP to Implement Black Box Testing on a Login Module

**Week 7: Version Control & Software Maintenance**

1. WAP to Demonstrate Git Commands for Version Control (commit, push, pull, branch)
2. WAP to Identify Different Types of Software Maintenance with Examples

**Week 8: Software Quality & Risk Management**

1. WAP to Demonstrate Software Quality Assurance Techniques with an Example
2. WAP to Perform Risk Analysis for a Banking System

**Week 9: Object-Oriented Design & Development**

1. WAP to Implement Encapsulation and Abstraction in Java/Python
2. WAP to Demonstrate Inheritance and Polymorphism using Java/Python

**Week 10: Web-Based Application Development**

1. WAP to Design a Simple Web Page for Online Registration using HTML & CSS
2. WAP to Validate a User Login Form using JavaScript

**Week 11: Software Project Management**

1. WAP to Develop a Project Plan for a Mobile Banking App
2. WAP to Implement Cost Estimation using COCOMO Model

**Week 12: Final Software Development Project**

1. WAP to Develop a Mini Software Project using SDLC Methodology (Library System, Online Store, etc.)
2. WAP to Prepare Final Software Documentation including SRS, Design, and Test Cases

**IoT & Sensor**

**Week 1: Introduction to IoT & Basic Setup**

1. WAP to Blink an LED using Arduino
2. WAP to Read and Display Analog Sensor Data (LDR/Temperature Sensor) using Arduino

**Week 2: Working with Sensors**

1. WAP to Interface a DHT11 Sensor and Display Temperature & Humidity on Serial Monitor
2. WAP to Measure Distance using Ultrasonic Sensor (HC-SR04) and Display on LCD

**Week 3: Actuators & Output Devices**

1. WAP to Control LED Brightness using PWM on Arduino
2. WAP to Interface a Servo Motor and Rotate to Specific Angles using Arduino

**Week 4: Communication with IoT Devices**

1. WAP to Send Data from ESP8266 to Serial Monitor (Wi-Fi Communication Test)
2. WAP to Control an LED using Web Server on ESP8266

**Week 5: Cloud Data Storage & Visualization**

1. WAP to Send Temperature and Humidity Data to ThingSpeak Cloud
2. WAP to Retrieve Data from ThingSpeak and Display on Serial Monitor

**Week 6: Home Automation**

1. WAP to Smart Home Automation using Relay & IoT
2. WAP to Control Electrical Appliances Remotely using Blynk App

**Week 7: Security & Monitoring Systems**

1. WAP to Detect Motion using PIR Sensor and Activate Alarm
2. WAP to RFID-based Smart Door Lock System

**Week 8: MQTT Protocol for IoT Communication**

1. WAP to Implement MQTT Publish and Subscribe using ESP8266
2. WAP to Control LED via MQTT Messages

**Week 9: Voice-Controlled IoT Applications**

1. WAP to Control LED using Google Assistant via IFTTT
2. WAP to Operate a Smart Fan using Voice Commands

**Week 10: Real-time Monitoring & Alerts**

1. WAP to Create an IoT-based Weather Monitoring System using Firebase
2. WAP to Send Email Alerts for High Temperature using IoT

**Week 11: Smart City Applications**

1. WAP to Implement a Smart Street Light System using LDR and IoT
2. WAP to Design an IoT-based Smart Parking System using IR Sensors

**Week 12: Smart Agriculture & Water Management**

1. WAP to Monitor Soil Moisture and Control Water Pump using IoT
2. WAP to Implement a Water Level Monitoring System with IoT

**Web Development Using PHP**

**Week 1: Introduction to PHP**

1. WAP to Print "Hello, World!" using PHP
2. WAP to Demonstrate PHP Variables and Data Types

**Week 2: Control Structures in PHP**

1. WAP to Check Whether a Number is Even or Odd using PHP
2. WAP to Find the Largest Among Three Numbers using PHP

**Week 3: Arrays and String Functions**

1. WAP to Demonstrate Different Array Operations in PHP (Indexed, Associative, and Multidimensional)
2. WAP to Perform String Operations like Concatenation, Substring, and String Length

**Week 4: Functions and Form Handling**

1. WAP to Create a User-Defined Function to Calculate Factorial of a Number
2. WAP to Create a Simple PHP Form and Display Submitted Data

**Week 5: Form Validation and File Handling**

1. WAP to Validate a User Registration Form (Name, Email, Password) using PHP
2. WAP to Read and Write Data to a Text File using PHP

**Week 6: Session and Cookie Management**

1. WAP to Store and Retrieve User Data using PHP Sessions
2. WAP to Set, Retrieve, and Delete Cookies in PHP

**Week 7: Database Connectivity with MySQL**

1. WAP to Connect PHP with MySQL Database
2. WAP to Insert, Update, and Delete Records in MySQL using PHP

**Week 8: User Authentication System**

1. WAP to Create a Simple Login System using PHP and MySQL
2. WAP to Implement User Logout using Sessions in PHP

**Week 9: File Upload and Image Handling**

1. WAP to Upload and Store an Image using PHP and MySQL
2. WAP to Display Uploaded Images in a Web Page using PHP

**Week 10: AJAX with PHP**

1. WAP to Implement AJAX in PHP for Real-Time Data Fetching
2. WAP to Create an Auto-Complete Search Box using AJAX and PHP

**Week 11: Email Handling and APIs in PHP**

1. WAP to Send Email using PHP Mail() Function
2. WAP to Integrate Third-Party APIs (e.g., Weather API) in PHP

**Week 12: Mini Project using PHP & MySQL**

1. WAP to Develop a Simple Blog System with CRUD Operations
2. WAP to Develop a Mini E-Commerce System with Product Listings and Cart Functionality

**AGGARWAL COLLEGE BALLABGARH**

**Class-M.C.A.-4th sem (Lab 2)**

**Web Development using .Net and Cyber Security**

**Week-1:**

* Write a program to print "Hello, World!"
* Write a program to find the Largest of Three Numbers.
* Write a program to find the Factorial of a Number.
* Write a program to generate a Multiplication Table for a Given Number.
* Write a program to Reverse a Number Using a While Loop**.**

**Week-2:**

* Write a program to Store and Print 5 Student Marks using One-Dimensional Array.
* Write a program to Perform Matrix Addition and Multiplication using Two-Dimensional Array.
* Write a program to check if a String is Palindrome.
* Write a program to implement Method Overloading and Overriding in C#.

**Week-3:**

* Write a program to implement Use of Abstract Classes and Interfaces.
* Write a program to implement Divide by Zero Exception Handling.
* Write a program to Handle Multiple Exceptions in a Single Try-Catch Block.
* Write a program to Demonstrate the Use of finally Block.

**Week-4:**

* Write a program to show the use of Polymorphism.
* Write a program to implement Multilevel Inheritance.
* Write a program to implement Hierarchical Inheritance.

**Week-5:**

* Write a program to Insert, Update, Delete Data Using ADO.NET.
* Write a program to view and filter data in ADO.NET.
* Write a program to create a Login System Using ADO.NET and SQL Server.

**Week-6:**

* Write a program to Design a login page with username and password fields using ASP.NET.
* Write a program to create a Registration Form using ASP.NET.
* Write a program to Create a Calculator Using ASP.NET Web Forms.

**Week-7:**

* Write a Python script to detect brute force attacks.
* Implement a simple key logger in Python.

**Week-8:**

* Create a script to monitor system processes for anomalies.
* Implement the Caesar cipher and Vigenère cipher.
* Develop a Python script for Playfair cipher encryption/decryption.

**Week-9:**

* Develop a digital signature verification program.
* Analyze network traffic using Wireshark and generate a report.

**Week-10:**

* Simulate a Man-in-the-Middle (MITM) attack using ARP spoofing.
* Develop a SQL Injection detection tool.
* Write a Python script to detect ransom ware signatures.

**Week-11:**

* Implement a simple block chain in Python.
* Develop a secure authentication system.

**Week-12:**

* Develop a file integrity checker for detecting malware infections.
* Develop a SIEM (Security Information and Event Management) tool prototype.