HANDS ON EXERCISE WEEK-8

1.Write a Java program that logs application version information and could be tracked using Git. Demonstrate how Git can be used to commit and retrieve version history for this file.

**Program (VersionLogger.java):**

import java.time.LocalDateTime;

public class VersionLogger {

public static void main(String[] args) {

String version = "v1.0.0";

String commitHash = "abc1234"; // Simulate Git commit hash

LocalDateTime buildTime = LocalDateTime.now();

System.out.println("Application Version: " + version);

System.out.println("Commit Hash: " + commitHash);

System.out.println("Build Time: " + buildTime);

}

}  
2.Write a Java program with a simple unit test that can be run automatically in a CI/CD pipeline (e.g., GitHub Actions, Jenkins).

**Program (Calculator.java):**

public class Calculator {

public int add(int a, int b) {

return a + b;

}

}

**Test (CalculatorTest.java using JUnit):**

import static org.junit.Assert.\*;

import org.junit.Test;

public class CalculatorTest {

@Test

public void testAdd() {

Calculator calc = new Calculator();

assertEquals(5, calc.add(2, 3));

}

}  
3.Write a Java program that reads configuration from an external .properties file, simulating how DevOps manages environment configurations.

**config.properties:**

app.name=DevOpsDemo

app.env=production

**Program (ConfigReader.java):**

import java.io.FileInputStream;

import java.io.IOException;

import java.util.Properties;

public class ConfigReader {

public static void main(String[] args) {

Properties props = new Properties();

try (FileInputStream fis = new FileInputStream("config.properties")) {

props.load(fis);

System.out.println("Application Name: " + props.getProperty("app.name"));

System.out.println("Environment: " + props.getProperty("app.env"));

} catch (IOException e) {

e.printStackTrace();

}

}

}  
4.Write a Java program that can be packaged in a Docker container and run anywhere.

**Program (HelloDocker.java):**

public class HelloDocker {

public static void main(String[] args) {

System.out.println("Hello from Java inside Docker!");

}

}

**Dockerfile:**

FROM openjdk:17

COPY HelloDocker.java /app/

WORKDIR /app

RUN javac HelloDocker.java

5.Write a Java program that simulates uploading a file to a cloud service (AWS S3-like example, without SDK for simplicity).

**Program (CloudUploader.java):**

import java.io.File;

public class CloudUploader {

public static void main(String[] args) {

File file = new File("sample.txt");

if (file.exists()) {

System.out.println("Uploading " + file.getName() + " to cloud...");

System.out.println("Upload successful!");

} else {

System.out.println("File not found.");

}

}

}

6.Write a Java program that simulates sending a prompt to a GenAI API and receiving a response.

**Program (GenAISimulator.java):**

import java.util.Scanner;

public class GenAISimulator {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter your prompt: ");

String prompt = sc.nextLine();

// Simulated AI response

String response = "AI Response to your prompt: " + prompt.toUpperCase();

System.out.println(response);

}

}