1. **Define an interface “Operation” which has methods area(),volume(). Define a constant Plhaving a value 3.142. Create a class circle (member — radius), cylinder (members — radius, height) which implements this interface. Calculate and display the area and volume.**

interface Operation {

double PI = 3.142; // Constant PI

double area();

double volume ();

}

class Circlel implements Operation {

private double radius;

public Circlel (double radius) {

this.radius = radius;

}

@Override

public double area() {

return PI \* radius \* radius;

}

@Override

public doublevolume() {

return 0;

}

}

class Cylinder implements Operation {

private double radius;

private double height;

public Cylinder(double radius, double height) {

this.radius = radius;

this.height = height;

}

@Override

public double area() {

return 2 \* PI \* radius \* (radius + height);

}

@Override

public double volume() {

return PI \* radius \* radius \* height;

}

}

public class Main 20 {

public static void main(String[] args) {

Circlel circle = new Circlel(5); // Radius = 5

System.out.printlin("Circle Area: " + circle.area());

System. out.printin("Circle Volume: " + circle.volume());

Cylinder cylinder = new Cylinder(5, 10); // Radius = 5, Height = 10

System. out.printin("Cylinder Area: " + cylinder.area());

System. out.printin("Cylinder Volume: " + cylinder.volume());

}

}

**2. Write a Java program to create a super class Employee (members — name, salary). Derive a subclass**

**as Developer (member — projectname). Derive a sub-class Programmer (member —**

**proglanguage) from Developer. Create object of Developer and display the details of it. Implement**

**this multilevel inheritance with appropriate constructor and methods**class Employee21 {

String name;

double salary;

public Employee21(String name, double salary) {

this.name = name;

this.salary = salary;

}

public void displayEmployeeDetails() {

System.out.println("Name: " + name);

System.out.printin("Salary: " + salary);

}

}

class Developer extends Employee21 {

String projectName;

public Developer(String name, double salary, String projectName) {

super(name, salary); // Calling the constructor of Employee class

this.projectName = projectName;

}

public void displayDeveloperDetails() {

displayEmployeeDetails();

System.out.printin("Project Name: " + projectName);

}

}

class Programmer extends Developer {

String progLanguage;

public Programmer (String name, double salary, String projectName, String progLanguage)

super(name, salary, projectName);

this.progLanguage = progLanguage;

}

public void displayProgrammerDetails() {

displayDeveloperDetails();

System.out.printin("Programming Language: " + progLanguage);

}

}

public class Main\_19 {

public static void main(String[] args)

Programmer programmer = new Programmer ("John Doe", 80000, "AI Project", "Java");

programmer .displayProgrammerDetails();

}

}

3**. Write a program for multilevel inheritance such that country is inherited from continent. State is**

**inherited from country. Display the place, state, country and continent.**

class Continent {

String continentName;

public Continent (String continentName) {

this.continentName = continentName;

}

public void displayContinent() {

System.out.printin("Continent: " + continentName) ;

}

}

class Country extends Continent {

String countryName;

public Country(String continentName, String countryName) {

super(continentName);

this.countryName = countryName;

}

public void displayCountry() {

displayContinent();

System.out.println("Country: " + countryName);

}

}

class State extends Country {

String stateName;

public State (String continentName, String countryName, String stateName) {

super(continentName, countryName);

this.stateName = stateName;

}

public void displayState() {

displayCountry();

System.out.printlin("State: " + stateName);

}

}

public class Main 21 {

publie static void main(String[] args) {

State state = new State("Asia", "India", "Maharashtra") ;

state.displayState();

}

}

**Assign 5**

**1.Write a java program that displays the number of characters, lines and words of a file.**

import java.io.BufferedReader;

import java.io.FileReader;

import java.io.I0Exception;

public class FileStatistics {

public static void main(String[] args) {

String filePath = "C:\\Users\\Kirti\\eclipse-workspace\\NGC\\practice programs\\sre\\simple\\sample.text";

int linecount = 0;

int wordcount O;

int charcount = 0;

try (BufferedReader reader = new BufferedRead(enrew FileReader(filePath))) {

String line;

while ((line = reader.readLine()) != null)

linecountt++;

charcount += line.length();

String[] words = line.split("\\st");

wordCount += words. length;

}

System.out.println("Number of Lines: " + lineCount);

System.out.println("Number of Words: " + wordCount);

System.out.printin("Number of Characters: " + charCount);

} catch (IOException =) {

System.out.printin("An error occurred while reading the file: " + e.getMessage());

}

}}

**2 .Write a program that accept user name and password and throws an exception “invalid login” if**

**they do not match.**

**Create a class LoginValidation.**

import java.util.Scanner;

class InvalidLoginException extends Exception {

public InvalidLoginExcep(tStiroinng message) {

super (message) ;

} }

public class LoginValidation {

public static void main(String[] args) {

String correctUsername = "admin";

String correctPassword = "password123";

Scanner scanner = new Scanner (System.in);

try {

System.out.print("Enter username: ");

String username = scanner.nextLine();

System.out.print ("Enter password: ");

String password = scanner.nextLine();

if (!username.equals(correctUsername) || !password.equals(correctPassword)) {

throw new InvalidLoginException ("Invalid login");

}

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

} catch (InvalidLoginException e) {

System.out.println("Exception: " + e.getMessage());

} finally {

scanner.close();

} }

}

**1.Write a java program to accept details of n customers (c\_id, cname, address,**

**mobile\_no) from user and store it in a file (Use DataOutputStream class). Display**

**the details of customers by reading it from file.(Use DatalnputStream class)**

import java.io.\*;

import java.util.Scanner;

public class CustomerDetails {

public static void main(String[] args) {

Scanner scanner = new Scanner (System.in);

System.out.print ("Enter the number of customers: ");

int n = scanner.nextiInt();

scanner.nextLine();

String fileName = "customers.dat";

try (DataOutputstream dos = new DataOutputStream(new FileOutputstream(fileName))) {

for (int i = 0; i <n; i++) {

System.out.printin("Enter details for customer "+ (i + 1) + ":");

System. out.print ("Customer ID: ");

int c\_id = scanner.nextInt();

scanner.nextLine();

System. out.print ("Customer Name: ");

String cname = scanner-.nextLine();

System. out.print ("Address: ");

String address = scanner.nextLine();

System.out.print ("Mobile Number: ");

String mobilen o = scanner.nextLine();

dos.writeInt(c\_id);

dos.writeUTF(cname) ;

dos.writeUTF (address) ;

dos.writeUTF (mobile no);

}

} catch (IOException e) {

System.out.println("Error writing to file: "+ e.getMessage());

}

try (DataInputStream dis = new DataInputStream(new FileInputStream(fileName))) {

System. out.printlin("\nCustomer Details:");

while (dis.available() > 0) {

int c\_id = dis.readInt();

String cname = dis.readUTF();

String address = dis.readUTF();

String mobile \_no = dis.readUTF();

System.out.println("Customer ID: " + c\_id);

System.out.println ("Customer Name: " + cname);

System.out.println("Address: " + address);

System.out.println("Mobile Number: " + mobile \_no);

System. out.println();

}

}

catch (IOException e) {

System.out.println ("Error reading from file: " + e.getMessage());

}

}

}

**2 .B Write a program to read a text file “sample.txt” and display the contents of a file in**

**reverse order and also original contents change the case (display in upper case).**

import java.io.\*;

public class FileManipulation {

public static void main(String[] args) {

String fileName = "C:\\Users\\Kirti\\eclipse-workspace\\NGC\\practice programs\\src\\simple\\uppercase.txt";

try {

System.out.println("Original Content in Uppercase:");

BufferedReader br = new BufferedRead(enrew FileReader (fileName) );

StringBuilder reverseContent = new StringBuilde(r);

String line;

while ((line = br.readLine()) != null) {

System. out.println(line.toUpperCase());

reverseContent.insert(0, line + "\n");

}

br.close();

System.out.printin("\nContent in Reverse Order:");

System. out.println(reverseContent.tostring()) +

} catch (IOException e) {

System.out.printin("Error: " + e.getMessage());

}}

}

**Assign 6**

**1.Write a Java program to create a basic window using AWT.**

import java.awt.\*;

public class BasicWindow{

public static void main(String[] args) {

Frame frame = new Frame("Basic Window");

frame.setSi(z40e0 , 300);

frame.setVisible (true);

}

}

**2.Write a Java program to create a Button using AWT.**

import java.awt.\*;

public class ButtonExample {

public static void main(String[] args) {

Frame frame = new Frame ("Button Example");

Button button = new Button("Click Me");

button.setBounds(100, 100, 80, 30);

frame.add (button);

frame.setSize(300, 200);

frame.setLayout (null);

frame .setVisi(btrluee) ;

}

}

**3.Write a Java program to create a button using AWT and demonstrate handling its click event with**

**an Event Listener**.

import java.awt.\*;[]

public class ButtonClick {

public static void main(String[] args) {

Frame frame = new Frame ("Button Click Example");

Button button = new Button("Click Me");

Label label = new Label ();

button.setBounds(100, 100, 80, 30);

label.setBounds(100, 150, 200, 30);

button. addActionListen(enrew ActionListener() {

public void actionPerformed(ActionEvent e&) {

label.setText ("Button Clicked!");

} });

frame.add (button);

frame.add(label);

frame.setSi(z30e0 , 300);

frame.setLay(noulul)t ;

frame.setVisible (true);

}

}

**4. Write a Java program using AWT to create a window with the following functionalities:**

1. A Label prompting the user to enter their name.

2. A TextField to input the name.

3. A Submit button that, when clicked, displays a personalized greeting message on the Label.

4. Ensure proper window handling to close the application

import java.awt.\*;

import java.awt.event.\*;

public class LabelAndTextFieldExample {

public static void main(String[] args) {

Frame frame = new Frame("Label, TextField, and Button Example");

Label label = new Label("Enter your name:");

label.setBounds(50, 50, 150, 30); // x, y, width, height

TextField textField = new TextField();

textField.setBounds(50, 100, 200, 30); // x, y, width, height

Button button = new Button ("Submit");

button.setBounds(50, 150, 80, 30); // x, y, width, height

button. addActionListen(enrew ActionListener() {

public void actionPerformed(ActionEvent e) {

String input = textField.getText();

label.setText("Hello, "+ input + "!");

}

frame.add(label);

frame.add(textField);

frame-add (button) ;|

frame.setSize(400, 250);

frame.setLayout (null);

frame.setVisible(true);

frame.addWindowListen(enrew WindowAdapter() {

public void windowClosing(WindowEvent e) {

frame.dispose();

}

}