

## Slip 11

**Q1. Define an interface `Operation`, and calculate the volume of a cylinder.**

- **Program:**

```
interface Operation {
    double PI = 3.142;
    double volume();
}
class Cylinder implements Operation {
    private double radius, height;
    public Cylinder(double radius, double height) {
        this.radius = radius;
        this.height = height;
    }
    public double volume() {
        return PI * radius * radius * height;
    }
}
public class MainCylinder {
    public static void main(String[] args) {
        Cylinder cylinder = new Cylinder(5, 10);
        System.out.println("Volume: " + cylinder.volume());
    }
}
```

**Q2. Accept username and password; throw `InvalidPasswordException` if not the same.**

- **Program:**

```
import java.util.Scanner;
class InvalidPasswordException extends Exception {
    public InvalidPasswordException(String msg) {
        super(msg);
    }
}
public class LoginSystem {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter Username: ");
        String username = sc.next();
        System.out.print("Enter Password: ");
        String password = sc.next();
        try {
            if (!username.equals(password))
```

```

        throw new InvalidPasswordException("Invalid Password! Username and Password
must be the same.");
        System.out.println("Login successful!");
    } catch (InvalidPasswordException e) {
        System.out.println(e.getMessage());
    }
}
}
}

```

---

## Slip 12

**Q1. Implement inheritance with **Department** extending **College**.**

- **Program:**

```

class College {
    int cno;
    String cname, caddr;
    public College(int cno, String cname, String caddr) {
        this.cno = cno;
        this.cname = cname;
        this.caddr = caddr;
    }
    public void displayCollegeDetails() {
        System.out.println("College: " + cname + ", Address: " + caddr);
    }
}

class Department extends College {
    int dno;
    String dname;
    public Department(int cno, String cname, String caddr, int dno, String dname) {
        super(cno, cname, caddr);
        this.dno = dno;
        this.dname = dname;
    }
    public void displayDetails() {
        displayCollegeDetails();
        System.out.println("Department: " + dname);
    }
}

public class CollegeDemo {
    public static void main(String[] args) {
        Department dept = new Department(101, "ABC College", "XYZ Street", 501, "Computer
Science");
        dept.displayDetails();
    }
}

```

**Q2. Write a program that demonstrates simple aggregation.**

- **Program:**

```
class Student {
    String name;
    int rollNo;
    Address addr;

    public Student(String name, int rollNo, Address addr) {
        this.name = name;
        this.rollNo = rollNo;
        this.addr = addr;
    }

    public void display() {
        System.out.println("Name: " + name + ", Roll No: " + rollNo);
        addr.displayAddress();
    }
}

class Address {
    String city, state;
    public Address(String city, String state) {
        this.city = city;
        this.state = state;
    }

    public void displayAddress() {
        System.out.println("City: " + city + ", State: " + state);
    }
}

public class AggregationDemo {
    public static void main(String[] args) {
        Address addr = new Address("Mumbai", "Maharashtra");
        Student student = new Student("John", 101, addr);
        student.display();
    }
}
```

---

## Slip 13

**Q1. Write a program to count words and lines from a file.**

- **Program:**

```
import java.io.*;
public class FileWordLineCount {
    public static void main(String[] args) throws IOException {
        if (args.length != 1) {
            System.out.println("Usage: java FileWordLineCount <file-name>");
            return;
        }
        File file = new File(args[0]);
        if (!file.exists()) {
            System.out.println("File does not exist.");
            return;
        }
        int wordCount = 0, lineCount = 0;
        try (BufferedReader br = new BufferedReader(new FileReader(file))) {
            String line;
            while ((line = br.readLine()) != null) {
                lineCount++;
                wordCount += line.split("\\s+").length;
            }
        }
        System.out.println("Lines: " + lineCount + ", Words: " + wordCount);
    }
}
```

## Q2. Write a program to display current system date and time in different formats.

- **Program:**

```
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.TimeZone;

public class DateTimeFormats {
    public static void main(String[] args) {
        Date currentDate = new Date();

        SimpleDateFormat format1 = new SimpleDateFormat("dd/MM/yyyy");
        System.out.println("Date (DD/MM/YYYY): " + format1.format(currentDate));

        SimpleDateFormat format2 = new SimpleDateFormat("MM-dd-yyyy");
        System.out.println("Date (MM-DD-YYYY): " + format2.format(currentDate));

        SimpleDateFormat format3 = new SimpleDateFormat("EEEE MMMM dd yyyy");
        System.out.println("Date (Day Month DD YYYY): " + format3.format(currentDate));
    }
}
```

```

SimpleDateFormat format4 = new SimpleDateFormat("EEE MMMM dd HH:mm:ss z
yyyy");
System.out.println("Date and Time (with timezone): " + format4.format(currentDate));

SimpleDateFormat format5 = new SimpleDateFormat("dd/MM/yy hh:mm:ss a Z");
format5.setTimeZone(TimeZone.getTimeZone("Asia/Kolkata"));
System.out.println("Date and Time (12-hour format with timezone): " +
format5.format(currentDate));
}
}

```

---

## Slip 14

**Q1. Check if a number is prime, raise exception for zero.**

- **Program:**

```

import java.util.Scanner;
class ZeroNumberException extends Exception {
    public ZeroNumberException(String msg) {
        super(msg);
    }
}
public class PrimeChecker {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter number: ");
        int number = sc.nextInt();
        try {
            if (number == 0)
                throw new ZeroNumberException("Number is zero!");
            if (isPrime(number))
                System.out.println(number + " is prime.");
            else
                System.out.println(number + " is not prime.");
        } catch (ZeroNumberException e) {
            System.out.println(e.getMessage());
        }
    }
    public static boolean isPrime(int n) {
        if (n <= 1) return false;
        for (int i = 2; i <= Math.sqrt(n); i++)
            if (n % i == 0) return false;
        return true;
    }
}

```

**Q2. Write a program using packages for SYMarks and TYMarks and calculate grades.**

- **Program:**

```
// Package SY
package SY;
public class SYMarks {
    public int computerTotal, mathsTotal, electronicsTotal;
    public SYMarks(int computerTotal, int mathsTotal, int electronicsTotal) {
        this.computerTotal = computerTotal;
        this.mathsTotal = mathsTotal;
        this.electronicsTotal = electronicsTotal;
    }
}

// Package TY
package TY;
public class TYMarks {
    public int theory, practicals;
    public TYMarks(int theory, int practicals) {
        this.theory = theory;
        this.practicals = practicals;
    }
}

// Main Program
import SY.SYMarks;
import TY.TYMarks;
public class StudentGrades {
    public static void main(String[] args) {
        SYMarks syMarks = new SYMarks(80, 75, 70);
        TYMarks tyMarks = new TYMarks(85, 90);
        int totalMarks = syMarks.computerTotal + tyMarks.theory + tyMarks.practicals;
        String grade = calculateGrade(totalMarks);
        System.out.println("Total Marks: " + totalMarks + ", Grade: " + grade);
    }
    public static String calculateGrade(int total) {
        if (total >= 70) return "A";
        if (total >= 60) return "B";
        if (total >= 50) return "C";
        return "Fail";
    }
}
```

## Slip 15

**Q1. Write a program to copy contents from one file to another.**

- **Program:**

```

import java.io.*;
public class FileCopy {
    public static void main(String[] args) throws IOException {
        BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));
        System.out.print("Enter source file: ");
        String source = reader.readLine();
        System.out.print("Enter destination file: ");
        String destination = reader.readLine();
        try (BufferedReader br = new BufferedReader(new FileReader(source));
            PrintWriter pw = new PrintWriter(new FileWriter(destination))) {
            String line;
            while ((line = br.readLine()) != null) {
                pw.println(line);
            }
            System.out.println("File copied successfully.");
        }
    }
}

```

**Q2. Implement a class **Account** with derived classes **SavingAccount** and **AccountDetail**.**

- **Program:**

```

class Account {
    String custName, accNo;
    public Account(String custName, String accNo) {
        this.custName = custName;
        this.accNo = accNo;
    }
    public void display() {
        System.out.println("Customer: " + custName + ", Account No: " + accNo);
    }
}

class SavingAccount extends Account {
    double savingBal, minBal;
    public SavingAccount(String custName, String accNo, double savingBal, double minBal) {
        super(custName, accNo);
        this.savingBal = savingBal;
        this.minBal = minBal;
    }
}

class AccountDetail extends SavingAccount {
    double depositAmt, withdrawalAmt;
    public AccountDetail(String custName, String accNo, double savingBal, double minBal,
        double depositAmt, double withdrawalAmt) {
        super(custName, accNo, savingBal, minBal);
    }
}

```

```

        this.depositAmt = depositAmt;
        this.withdrawalAmt = withdrawalAmt;
    }
    @Override
    public void display() {
        super.display();
        System.out.println("Saving Balance: " + savingBal + ", Min Balance: " + minBal);
        System.out.println("Deposit: " + depositAmt + ", Withdrawal: " + withdrawalAmt);
    }
}

public class AccountMain {
    public static void main(String[] args) {
        AccountDetail accDetail = new AccountDetail("John Doe", "123456", 5000, 1000, 2000,
500);
        accDetail.display();
    }
}

```

---

## Slip 16

**Q1. Write a program using a lambda expression to calculate the square of a number.**

- **Program:**

```

import java.util.Scanner;
interface Square {
    int calculate(int number);
}
public class SquareCalculator {
    public static void main(String[] args) {
        Square square = (num) -> num * num;
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int number = sc.nextInt();
        System.out.println("Square: " + square.calculate(number));
    }
}

```

**Q2. Create an AWT program that accepts a name and displays a greeting message.**

- **Program:**

```

import java.awt.*;
import java.awt.event.*;

```



```

public class AWTEExample extends Frame implements ActionListener {
    Label label;
    TextField textField;
    Button button;

    public AWTEExample() {
        label = new Label("Enter your name:");
        textField = new TextField(20);
        button = new Button("Submit");
        setLayout(new FlowLayout());
        add(label);
        add(textField);
        add(button);
        button.addActionListener(this);
        setTitle("AWT Example");
        setSize(300, 150);
        setVisible(true);
        addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent we) {
                System.exit(0);
            }
        });
    }

    public void actionPerformed(ActionEvent e) {
        String name = textField.getText();
        System.out.println("Hello, " + name + "!");
    }

    public static void main(String[] args) {
        new AWTEExample();
    }
}

```

---

## Slip 17

**Q1. Write a program with inheritance involving **Customer**, **Depositor**, and **Borrower**.**

- **Program:**

```

import java.util.Scanner;
class Customer {
    String name, phoneNumber;
    void readCustomerDetails(Scanner sc) {
        System.out.print("Enter Customer Name: ");
        name = sc.nextLine();
        System.out.print("Enter Phone Number: ");
    }
}

```

```

        phoneNumber = sc.nextLine();
    }
    void displayCustomerDetails() {
        System.out.println("Name: " + name + ", Phone: " + phoneNumber);
    }
}

class Depositor extends Customer {
    String accNo;
    double balance;
    void readDepositorDetails(Scanner sc) {
        readCustomerDetails(sc);
        System.out.print("Enter Account Number: ");
        accNo = sc.nextLine();
        System.out.print("Enter Balance: ");
        balance = sc.nextDouble();
    }
    void displayDepositorDetails() {
        displayCustomerDetails();
        System.out.println("Account: " + accNo + ", Balance: " + balance);
    }
}

class Borrower extends Depositor {
    String loanNo;
    double loanAmt;
    void readBorrowerDetails(Scanner sc) {
        readDepositorDetails(sc);
        System.out.print("Enter Loan Number: ");
        loanNo = sc.next();
        System.out.print("Enter Loan Amount: ");
        loanAmt = sc.nextDouble();
    }
    void displayBorrowerDetails() {
        displayDepositorDetails();
        System.out.println("Loan No: " + loanNo + ", Loan Amount: " + loanAmt);
    }
}

public class CustomerManagement {
    public static void main(String[] args) {
        Borrower borrower = new Borrower();
        Scanner sc = new Scanner(System.in);
        borrower.readBorrowerDetails(sc);
        borrower.displayBorrowerDetails();
    }
}

```

## Q2. Write a GUI application that concatenates two strings or reverses the first string.

- **Program:**

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

public class StringManipulator extends JFrame implements ActionListener {
    JTextField textBox1, textBox2, resultBox;
    JButton concatButton, reverseButton;

    public StringManipulator() {
        textBox1 = new JTextField(15);
        textBox2 = new JTextField(15);
        resultBox = new JTextField(15);
        resultBox.setEditable(false);
        concatButton = new JButton("Concatenate");
        reverseButton = new JButton("Reverse");
        setLayout(new FlowLayout());
        add(new JLabel("String 1:"));
        add(textBox1);
        add(new JLabel("String 2:"));
        add(textBox2);
        add(concatButton);
        add(reverseButton);
        add(new JLabel("Result:"));
        add(resultBox);
        concatButton.addActionListener(this);
        reverseButton.addActionListener(this);
        setTitle("String Manipulator");
        setSize(300, 200);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setVisible(true);
    }

    public void actionPerformed(ActionEvent e) {
        if (e.getSource() == concatButton) {
            resultBox.setText(textBox1.getText() + textBox2.getText());
        } else if (e.getSource() == reverseButton) {
            resultBox.setText(new StringBuilder(textBox1.getText()).reverse().toString());
        }
    }

    public static void main(String[] args) {
        new StringManipulator();
    }
}
```

---

## Slip 18

**Q1. Write a program to demonstrate **BorderLayout** using AWT/Swing.**

- **Program:**

```
import javax.swing.*;
import java.awt.*;

public class BorderLayoutExample extends JFrame {
    public BorderLayoutExample() {
        setTitle("Border Layout Example");
        setLayout(new BorderLayout());
        add(new JButton("North"), BorderLayout.NORTH);
        add(new JButton("South"), BorderLayout.SOUTH);
        add(new JButton("East"), BorderLayout.EAST);
        add(new JButton("West"), BorderLayout.WEST);
        add(new JButton("Center"), BorderLayout.CENTER);
        setSize(400, 300);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setVisible(true);
    }

    public static void main(String[] args) {
        new BorderLayoutExample();
    }
}
```

**Q2. Write a program to calculate the batting average of cricket players and sort them by average.**

- **Program:**

```
import java.util.Arrays;
import java.util.Scanner;

class CricketPlayer {
    String name;
    int noOfInnings, noOfTimesNotOut, totalRuns;
    double batAvg;

    public CricketPlayer(String name, int noOfInnings, int noOfTimesNotOut, int totalRuns) {
        this.name = name;
        this.noOfInnings = noOfInnings;
        this.noOfTimesNotOut = noOfTimesNotOut;
        this.totalRuns = totalRuns;
    }
}
```

```

        this.batAvg = calculateAverage();
    }

    private double calculateAverage() {
        if (noOfInnings - noOfTimesNotOut > 0)
            return (double) totalRuns / (noOfInnings - noOfTimesNotOut);
        else
            return 0.0;
    }

    public static void sortPlayers(CricketPlayer[] players) {
        Arrays.sort(players, (a, b) -> Double.compare(b.batAvg, a.batAvg));
    }

    public void displayDetails() {
        System.out.printf("Name: %s, Innings: %d, Not Out: %d, Total Runs: %d, Batting Average: %.2f%n",
            name, noOfInnings, noOfTimesNotOut, totalRuns, batAvg);
    }
}

public class CricketPlayerManager {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter number of players: ");
        int n = sc.nextInt();
        CricketPlayer[] players = new CricketPlayer[n];

        for (int i = 0; i < n; i++) {
            System.out.printf("Enter details for Player %d:%n", i + 1);
            System.out.print("Name: ");
            String name = sc.next();
            System.out.print("Innings: ");
            int innings = sc.nextInt();
            System.out.print("Not Out: ");
            int notOut = sc.nextInt();
            System.out.print("Total Runs: ");
            int runs = sc.nextInt();
            players[i] = new CricketPlayer(name, innings, notOut, runs);
        }

        CricketPlayer.sortPlayers(players);
        System.out.println("\nSorted Player Details:");
        for (CricketPlayer player : players) {
            player.displayDetails();
        }
    }
}

```

---

## Slip 19

**Q1. Write a program to calculate the sum of primary and secondary diagonal elements of a square matrix.**

- **Program:**

```
import java.util.Scanner;

public class DiagonalSum {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter size of matrix (n x n): ");
        int n = sc.nextInt();
        int[][] matrix = new int[n][n];

        System.out.println("Enter matrix elements:");
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < n; j++) {
                System.out.printf("Element [%d][%d]: ", i, j);
                matrix[i][j] = sc.nextInt();
            }
        }

        int primarySum = 0, secondarySum = 0;
        for (int i = 0; i < n; i++) {
            primarySum += matrix[i][i];
            secondarySum += matrix[i][n - 1 - i];
        }

        System.out.println("Primary Diagonal Sum: " + primarySum);
        System.out.println("Secondary Diagonal Sum: " + secondarySum);

        if (n % 2 == 1) {
            int center = matrix[n / 2][n / 2];
            System.out.println("Total Diagonal Sum: " + (primarySum + secondarySum - center));
        } else {
            System.out.println("Total Diagonal Sum: " + (primarySum + secondarySum));
        }
    }
}
```

**Q2. Write a program using JComboBox to display TYBSc subjects and show the selected subject.**

- **Program:**

```

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

public class SubjectSelector extends JFrame implements ActionListener {
    JComboBox<String> subjectComboBox;
    JTextField selectedSubjectField;

    public SubjectSelector() {
        setTitle("T.Y.B.Sc. Subjects");
        String[] subjects = {"Data Structures", "Database Management", "Software Engineering",
"Computer Networks", "Web Technologies", "Operating Systems", "Theory of
Computation"};
        subjectComboBox = new JComboBox<>(subjects);
        selectedSubjectField = new JTextField(20);
        selectedSubjectField.setEditable(false);
        JButton showButton = new JButton("Show Selected Subject");

        setLayout(new FlowLayout());
        add(new JLabel("Select Subject:"));
        add(subjectComboBox);
        add(showButton);
        add(new JLabel("Selected Subject:"));
        add(selectedSubjectField);

        showButton.addActionListener(this);
        setSize(300, 200);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setVisible(true);
    }

    @Override
    public void actionPerformed(ActionEvent e) {
        String selectedSubject = (String) subjectComboBox.getSelectedItem();
        selectedSubjectField.setText(selectedSubject);
    }

    public static void main(String[] args) {
        new SubjectSelector();
    }
}

```

---

## Slip 20

**Q1. Write a program demonstrating multilevel inheritance with **Continent**, **Country**, and **State**.**

- **Program:**

```
class Continent {
    String name;
    public Continent(String name) {
        this.name = name;
    }
    public void displayContinent() {
        System.out.println("Continent: " + name);
    }
}

class Country extends Continent {
    String countryName;
    public Country(String continentName, String countryName) {
        super(continentName);
        this.countryName = countryName;
    }
    public void displayCountry() {
        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 displayDetails() {
        displayContinent();
        displayCountry();
        System.out.println("State: " + stateName);
    }
}

public class InheritanceExample {
    public static void main(String[] args) {
        State state = new State("North America", "United States", "California");
        state.displayDetails();
    }
}
```

**Q2. Write a package for **Operation** with two classes: **Addition** and **Maximum**.**

- **Addition.java (inside Operation package):**

```
package Operation;
```



```
public class Addition {  
    public int add(int a, int b) {  
        return a + b;  
    }  
  
    public float subtract(float a, float b) {  
        return a - b;  
    }  
}
```

- **Maximum.java (inside Operation package):**

```
package Operation;  
  
public class Maximum {  
    public void max(int a, int b) {  
        int maximum = (a > b) ? a : b;  
        System.out.println("Maximum: " + maximum);  
    }  
}
```

- **Main.java:**

```
import Operation.Addition;  
import Operation.Maximum;  
  
public class Main {  
    public static void main(String[] args) {  
        Addition addition = new Addition();  
        Maximum maximum = new Maximum();  
  
        int sum = addition.add(5, 10);  
        float difference = addition.subtract(15.5f, 5.2f);  
        System.out.println("Sum: " + sum);  
        System.out.println("Difference: " + difference);  
  
        maximum.max(10, 20);  
    }  
}
```