slip1_1.java

```java
//? prime number = slip1_1

public class slip1_1 {

    public static boolean isPrime(int n) {
        if (n < 2){
            return false;
        }
        for (int i = 2; i * i <= n; i++) {
            if (n % i == 0){
                return false;
            }
        }
        return true;
    }

    public static void main(String[] args) {
        for (String arg : args) {
            int n = Integer.parseInt(arg);
            if (isPrime(n)) {
                System.out.print(n + " ");
            }
        }
    }
}

// output =
// //? javac slip1_1.java
// //? java slip1_1 3 4 5 6 7 8 9
// //! ==========> 3 5 7
```

slip1_2.java

```java
//? Define an abstract class Staff with protected members id and name. Define a parameterized
//? constructor. Define one subclass OfficeStaff with member department. Create n objects of
//? OfficeStaff and display all details.

import java.util.Scanner;

abstract class Staff {
    protected int id;
    protected String name;

    public Staff(int id, String name) {
        this.id = id;
        this.name = name;
    }

    // Abstract method to be implemented by subclasses
    public abstract void displayDetails();
}

class OfficeStaff extends Staff {
    private String department;

    public OfficeStaff(int id, String name, String department) {
        super(id, name);
        this.department = department;
    }

    @Override
    public void displayDetails() {
        System.out.println("ID: " + id);
        System.out.println("Name: " + name);
        System.out.println("Department: " + department);
    }
}

public class slip1_2 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the number of Office Staff: ");
        int n = scanner.nextInt();
        scanner.nextLine(); // Consume newline

        OfficeStaff[] staffArray = new OfficeStaff[n];

        for (int i = 0; i < n; i++) {
            System.out.println("\nEnter details for Office Staff " + (i + 1) + ":");
            System.out.print("Enter ID: ");
            int id = scanner.nextInt();
            scanner.nextLine(); // Consume newline
            System.out.print("Enter Name: ");
            String name = scanner.nextLine();
```

```
            System.out.print("Enter Department: ");
            String department = scanner.nextLine();

            staffArray[i] = new OfficeStaff(id, name, department);
        }

        System.out.println("\n--- Office Staff Details ---");
        for (OfficeStaff staff : staffArray) {
            staff.displayDetails();
            System.out.println("----------------------------");
        }

        scanner.close();
    }
}

// output =
// Enter the number of Office Staff: 3

// Enter details for Office Staff 1:
// Enter ID: 1
// Enter Name: Rohit
// Enter Department: sale

// Enter details for Office Staff 2:
// Enter ID: 2
// Enter Name: viraj
// Enter Department: sale

// Enter details for Office Staff 3:
// Enter ID: 3
// Enter Name: ritesh
// Enter Department: submanager

// --- Office Staff Details ---
// ID: 1
// Name: Rohit
// Department: sale
// ----------------------------
// ID: 2
// Name: viraj
// Department: sale
// ----------------------------
// ID: 3
// Name: ritesh
// Department: submanager
// ----------------------------
```

slip3_1.java

```java
//? Write a program to accept ‘n’ name of cities from the user and sort them in ascending order.

import java.util.*;


public class slip3_1 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the number of cities: ");
        int n = scanner.nextInt();
        scanner.nextLine(); // Consume newline

        String[] cities = new String[n];
        for (int i = 0; i < n; i++) {
            System.out.print("Enter city name " + (i + 1) + ": ");
            cities[i] = scanner.nextLine();
        }

        // Sort the city names in ascending order
        Arrays.sort(cities);

        System.out.println("\nCities in Ascending Order:");
        for (String city : cities) {
            System.out.println(city);
        }

        scanner.close();
    }
}

// output =
// Enter the number of cities: 5
// Enter city name 1: pune
// Enter city name 2: mumbai
// Enter city name 3: new york
// Enter city name 4: delhi
// Enter city name 5: jaipur

// Cities in Ascending Order:
// delhi
// jaipur
// mumbai
// new york
// pune
```

slip3_2.java

```
//? Define a class patient (patient_name, patient_age, patient_oxy_level,patient_ HRCT _report).
//? Create an object of patient. Handle appropriate exception while patient oxygen level less
than
//? 95% and HRCT scan report greater than 10, then throw user defined Exception “Patient is
Covid
//? Positive(+) and Need to Hospitalized™ otherwise display its information.


// Custom Exception Class
class CovidPositiveException extends Exception {
    public CovidPositiveException(String message) {
        super(message);
    }
}

// Patient Class
class Patient {
    String name;
    int age;
    int oxyLevel;
    int hrctReport;

    public Patient(String name, int age, int oxyLevel, int hrctReport) {
        this.name = name;
        this.age = age;
        this.oxyLevel = oxyLevel;
        this.hrctReport = hrctReport;
    }

    public void checkHealth() throws CovidPositiveException {
        if (oxyLevel < 95 && hrctReport > 10) {
            throw new CovidPositiveException("Patient is Covid Positive(+) and Needs to be
Hospitalized™");
        } else {
            System.out.println("Name: " + name);
            System.out.println("Age: " + age);
            System.out.println("Oxygen Level: " + oxyLevel + "%");
            System.out.println("HRCT Report: " + hrctReport);
        }
    }
}

// Main Class
public class slip3_2 {
    public static void main(String[] args) {
        Patient patient = new Patient("John Doe", 45, 92, 12);

        try {
            patient.checkHealth();
        } catch (CovidPositiveException e) {
            System.out.println(e.getMessage());
        }
```

```
        }
    }

    // output =
    // Patient is Covid Positive(+) and Needs to be Hospitalized?
```

slip5_1.java

```
//? 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.


//! this program also in slip20_1


class Continent {
    String name;

    Continent(String name) {
        this.name = name;
    }
}

class Country extends Continent {
    String countryName;

    Country(String continentName, String countryName) {
        super(continentName);
        this.countryName = countryName;
    }
}

class State extends Country {
    String stateName;
    String placeName;

    State(String continentName, String countryName, String stateName , String placeName) {
        super(continentName, countryName);
        this.stateName = stateName;
        this.placeName = placeName;
    }

    void display() {
       System.out.println("continent Name : " + name);
       System.out.println("County Name : "+ countryName);
       System.out.println("State Name :"+ stateName);
       System.out.println("place name " + placeName);
    }
}

public class slip5_1 {
    public static void main(String[] args) {
        State s = new State("Asia", "India", "Maharashtra","pune");
        s.display();
    }
}

// output =
// continent Name : Asia
```

```
// County Name : India
// State Name :Maharashtra
// place name pune
```

slip5_2.java

```
//?    Write a menu driven program to perform the following operations on multidimensional array
//?    ie matrices :
//?    = Addition
//?    = Multiplication
//?    = Exit

import java.util.Scanner;

public class slip5_2 {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int choice;

        do {
            System.out.println("\nMatrix Operations Menu:");
            System.out.println("1. Addition");
            System.out.println("2. Multiplication");
            System.out.println("3. Exit");
            System.out.print("Enter your choice: ");
            choice = sc.nextInt();

            switch (choice) {
                case 1:
                    addMatrices(sc);
                    break;
                case 2:
                    multiplyMatrices(sc);
                    break;
                case 3:
                    System.out.println("Exiting the program.");
                    break;
                default:
                    System.out.println("Invalid choice. Please try again.");
            }
        } while (choice != 3);

        sc.close();
    }

    private static void addMatrices(Scanner sc) {
        System.out.print("Enter number of rows: ");
        int rows = sc.nextInt();
        System.out.print("Enter number of columns: ");
        int cols = sc.nextInt();

        int[][] matrixA = new int[rows][cols];
        int[][] matrixB = new int[rows][cols];
        int[][] sumMatrix = new int[rows][cols];

        System.out.println("Enter elements of Matrix A:");
```

```
        fillMatrix(matrixA, sc);

        System.out.println("Enter elements of Matrix B:");
        fillMatrix(matrixB, sc);

        // Adding matrices
        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < cols; j++) {
                sumMatrix[i][j] = matrixA[i][j] + matrixB[i][j];
            }
        }

        System.out.println("Result of Matrix Addition:");
        printMatrix(sumMatrix);
    }

    private static void multiplyMatrices(Scanner sc) {
        System.out.print("Enter number of rows for Matrix A: ");
        int rowsA = sc.nextInt();
        System.out.print("Enter number of columns for Matrix A (and rows for Matrix B): ");
        int colsA = sc.nextInt();

        System.out.print("Enter number of columns for Matrix B: ");
        int colsB = sc.nextInt();

        int[][] matrixA = new int[rowsA][colsA];
        int[][] matrixB = new int[colsA][colsB];
        int[][] productMatrix = new int[rowsA][colsB];

        System.out.println("Enter elements of Matrix A:");
        fillMatrix(matrixA, sc);

        System.out.println("Enter elements of Matrix B:");
        fillMatrix(matrixB, sc);

        // Multiplying matrices
        for (int i = 0; i < rowsA; i++) {
            for (int j = 0; j < colsB; j++) {
                productMatrix[i][j] = 0; // Initialize to 0
                for (int k = 0; k < colsA; k++) {
                    productMatrix[i][j] += matrixA[i][k] * matrixB[k][j];
                }
            }
        }

        System.out.println("Result of Matrix Multiplication:");
        printMatrix(productMatrix);
    }

    private static void fillMatrix(int[][] matrix, Scanner sc) {
        for (int i = 0; i < matrix.length; i++) {
            for (int j = 0; j < matrix[i].length; j++) {
                System.out.print("Element [" + i + "][" + j + "]: ");
                matrix[i][j] = sc.nextInt();
```

```
            }
        }
    }

    private static void printMatrix(int[][] matrix) {
        for (int[] row : matrix) {
            for (int element : row) {
                System.out.print(element + " ");
            }
            System.out.println();
        }
    }
}

// output =
// Matrix Operations Menu:
// 1. Addition
// 2. Multiplication
// 3. Exit
// Enter your choice: 1
// Enter number of rows: 3
// Enter number of columns: 3
// Enter elements of Matrix A:
// Element [0][0]: 1
// Element [0][1]: 2
// Element [0][2]: 3
// Element [1][0]: 4
// Element [1][1]: 5
// Element [1][2]: 6
// Element [2][0]: 7
// Element [2][1]: 8
// Element [2][2]: 9
// Enter elements of Matrix B:
// Element [0][0]: 1
// Element [0][1]: 2
// Element [0][2]: 3
// Element [1][0]: 4
// Element [1][1]: 5
// Element [1][2]: 6
// Element [2][0]: 7
// Element [2][1]: 8
// Element [2][2]: 9
// Result of Matrix Addition:
// 2 4 6
// 8 10 12
// 14 16 18

// Matrix Operations Menu:
// 1. Addition
// 2. Multiplication
// 3. Exit
// Enter your choice: 2
// Enter number of rows for Matrix A: 2
// Enter number of columns for Matrix A (and rows for Matrix B): 2
```

```
// Enter number of columns for Matrix B: 2
// Enter elements of Matrix A:
// Element [0][0]: 1
// Element [0][1]: 2
// Element [1][0]: 3
// Element [1][1]: 4
// Enter elements of Matrix B:
// Element [0][0]: 1
// Element [0][1]: 2
// Element [1][0]: 3
// Element [1][1]: 4
// Result of Matrix Multiplication:
// 7 10
// 15 22
```

slip6_1.java

```
//? Write a program to display the Employee(Empid, Empname, Empdesignation, Empsal) information using toString().

class Employee {
    private int empId;
    private String empName;
    private String empDesignation;
    private double empSal;

    // Constructor
    public Employee(int empId, String empName, String empDesignation, double empSal) {
        this.empId = empId;
        this.empName = empName;
        this.empDesignation = empDesignation;
        this.empSal = empSal;
    }

    // Override toString() method
    @Override
    public String toString() {
        return "Employee ID: " + empId + "\n" +
               "Employee Name: " + empName + "\n" +
               "Employee Designation: " + empDesignation + "\n" +
               "Employee Salary: $" + empSal;
    }
}

public class slip6_1 {
    public static void main(String[] args) {
        // Creating Employee objects
        Employee emp1 = new Employee(101, "Mohan prakash", "Software Engineer", 75000);
        Employee emp2 = new Employee(102, "Rohan pawar", "Project Manager", 90000);
        Employee emp3 = new Employee(103, "Viraj kabra", "QA Analyst", 65000);

        // Displaying employee information using toString()
        System.out.println("Employee Information:");
        System.out.println(emp1);
        System.out.println();
        System.out.println(emp2);
        System.out.println();
        System.out.println(emp3);
    }
}

// output =
// Employee Information:
// Employee ID: 101
// Employee Name: Mohan prakash
// Employee Designation: Software Engineer
// Employee Salary: $75000.0

// Employee ID: 102
```

```
// Employee Name: Rohan pawar
// Employee Designation: Project Manager
// Employee Salary: $90000.0

// Employee ID: 103
// Employee Name: Viraj kabra
// Employee Designation: QA Analyst
// Employee Salary: $65000.0
```

slip6_2.java

```
//? Create an abstract class “order” having members id, description. Create two subclasses
//? “PurchaseOrder” and “Sales Order” having members customer name and Vendor name
//? respectively. Definemethods accept and display in all cases. Create 3 objects each of
Purchase
//? Order and Sales Order and accept and display details.

import java.util.Scanner;

// Abstract class Order
abstract class Order {
    protected int id;
    protected String description;

    // Abstract methods
    public abstract void accept();
    public abstract void display();
}

// Subclass PurchaseOrder
class PurchaseOrder extends Order {
    private String vendorName;

    @Override
    public void accept() {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter Purchase Order ID: ");
        id = sc.nextInt();
        sc.nextLine(); // Consume newline
        System.out.print("Enter Description: ");
        description = sc.nextLine();
        System.out.print("Enter Vendor Name: ");
        vendorName = sc.nextLine();

    }

    @Override
    public void display() {
        System.out.println("Purchase Order Details:");
        System.out.println("ID: " + id);
        System.out.println("Description: " + description);
        System.out.println("Vendor Name: " + vendorName);
    }
}

// Subclass SalesOrder
class SalesOrder extends Order {
    private String customerName;

    @Override
    public void accept() {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter Sales Order ID: ");
```

```
        id = sc.nextInt();
        sc.nextLine(); // Consume newline
        System.out.print("Enter Description: ");
        description = sc.nextLine();
        System.out.print("Enter Customer Name: ");
        customerName = sc.nextLine();

    }

    @Override
    public void display() {
        System.out.println("Sales Order Details:");
        System.out.println("ID: " + id);
        System.out.println("Description: " + description);
        System.out.println("Customer Name: " + customerName);
    }
}

public class slip6_2 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        // Creating arrays for Purchase and Sales Orders
        PurchaseOrder[] purchaseOrders = new PurchaseOrder[3];
        SalesOrder[] salesOrders = new SalesOrder[3];

        // Accepting details for Purchase Orders
        for (int i = 0; i < 3; i++) {
            purchaseOrders[i] = new PurchaseOrder();
            System.out.println("\n--- Enter details for Purchase Order " + (i + 1) + " ---");
            purchaseOrders[i].accept();
        }

        // Accepting details for Sales Orders
        for (int i = 0; i < 3; i++) {
            salesOrders[i] = new SalesOrder();
            System.out.println("\n--- Enter details for Sales Order " + (i + 1) + " ---");
            salesOrders[i].accept();
        }

        // Displaying details of Purchase Orders
        System.out.println("\n--- Displaying Purchase Orders ---");
        for (PurchaseOrder po : purchaseOrders) {
            po.display();
            System.out.println();
        }

        // Displaying details of Sales Orders
        System.out.println("--- Displaying Sales Orders ---");
        for (SalesOrder so : salesOrders) {
            so.display();
            System.out.println();
        }
```

```
        sc.close(); // Close the sc to avoid resource leaks
    }
}

// output =
// --- Enter details for Purchase Order 1 ---
// Enter Purchase Order ID: 101
// Enter Description: toy
// Enter Vendor Name: dev

// --- Enter details for Purchase Order 2 ---
// Enter Purchase Order ID: 102
// Enter Description: doll
// Enter Vendor Name: rushi

// --- Enter details for Purchase Order 3 ---
// Enter Purchase Order ID: 103
// Enter Description: books
// Enter Vendor Name: rohit

// --- Enter details for Sales Order 1 ---
// Enter Sales Order ID: 201
// Enter Description: toy
// Enter Customer Name: rohan

// --- Enter details for Sales Order 2 ---
// Enter Sales Order ID: 202
// Enter Description: doll
// Enter Customer Name: netal

// --- Enter details for Sales Order 3 ---
// Enter Sales Order ID: 203
// Enter Description: books
// Enter Customer Name: atharv

// --- Displaying Purchase Orders ---
// Purchase Order Details:
// ID: 101
// Description: toy
// Vendor Name: dev

// Purchase Order Details:
// ID: 102
// Description: doll
// Vendor Name: rushi

// Purchase Order Details:
// ID: 103
// Description: books
// Vendor Name: rohit

// --- Displaying Sales Orders ---
// Sales Order Details:
// ID: 201
```

```
// Description: toy
// Customer Name: rohan

// Sales Order Details:
// ID: 202
// Description: doll
// Customer Name: netal

// Sales Order Details:
// ID: 203
// Description: books
// Customer Name: atharv
```

slip7_1.java

```
// Design a class for Bank. Bank Class should support following operations;
// a. Deposit a certain amount into an account
// b. Withdraw a certain amount from an account
// c. Return a Balance value specifying the amount with details

import java.util.Scanner;

class Bank {
    private String accountHolderName;
    private String accountNumber;
    private double balance;

    // Constructor
    public Bank(String accountHolderName, String accountNumber) {
        this.accountHolderName = accountHolderName;
        this.accountNumber = accountNumber;
        this.balance = 0.0; // Initial balance is set to 0
    }

    // Method to deposit money
    public void deposit(double amount) {
        if (amount > 0) {
            balance += amount;
            System.out.println("Successfully deposited: $" + amount);
        } else {
            System.out.println("Deposit amount must be positive.");
        }
    }

    // Method to withdraw money
    public void withdraw(double amount) {
        if (amount > 0 && amount <= balance) {
            balance -= amount;
            System.out.println("Successfully withdrew: $" + amount);
        } else if (amount > balance) {
            System.out.println("Insufficient funds for withdrawal.");
        } else {
            System.out.println("Withdrawal amount must be positive.");
        }
    }

    // Method to return the balance with details
    public String getBalance() {
        return "Account Holder: " + accountHolderName + "\n" +
               "Account Number: " + accountNumber + "\n" +
               "Current Balance: $" + balance;
    }
}

public class slip7_1 {
    public static void main(String[] args) {
```

```
        Scanner sc = new Scanner(System.in);

        // Create a bank account
        System.out.print("Enter Account Holder Name: ");
        String name = sc.nextLine();
        System.out.print("Enter Account Number: ");
        String accNumber = sc.nextLine();

        Bank bankAccount = new Bank(name, accNumber);

        int choice;

        do {
            System.out.println("\nBank Operations Menu:");
            System.out.println("1. Deposit");
            System.out.println("2. Withdraw");
            System.out.println("3. Check Balance");
            System.out.println("4. Exit");
            System.out.print("Enter your choice: ");
            choice = sc.nextInt();

            switch (choice) {
                case 1:
                    System.out.print("Enter amount to deposit: $");
                    double depositAmount = sc.nextDouble();
                    bankAccount.deposit(depositAmount);
                    break;

                case 2:
                    System.out.print("Enter amount to withdraw: $");
                    double withdrawAmount = sc.nextDouble();
                    bankAccount.withdraw(withdrawAmount);
                    break;

                case 3:
                    System.out.println("\n--- Account Balance ---");
                    System.out.println(bankAccount.getBalance());
                    break;

                case 4:
                    System.out.println("Exiting the program.");
                    break;

                default:
                    System.out.println("Invalid choice. Please try again.");
            }
        } while (choice != 4);

        sc.close(); // Close the sc to avoid resource leaks
    }
}

// output =
// Enter Account Holder Name: rohan
```

```
// Enter Account Number: 485485485485

// Bank Operations Menu:
// 1. Deposit
// 2. Withdraw
// 3. Check Balance
// 4. Exit
// Enter your choice: 1
// Enter amount to deposit: $10000
// Successfully deposited: $10000.0

// Bank Operations Menu:
// 1. Deposit
// 2. Withdraw
// 3. Check Balance
// 4. Exit
// Enter your choice: 1
// Enter amount to deposit: $50000
// Successfully deposited: $50000.0

// Bank Operations Menu:
// 1. Deposit
// 2. Withdraw
// 3. Check Balance
// 4. Exit
// Enter your choice: 2
// Enter amount to withdraw: $2000
// Successfully withdrew: $2000.0

// Bank Operations Menu:
// 1. Deposit
// 2. Withdraw
// 3. Check Balance
// 4. Exit
// Enter your choice: 3

// --- Account Balance ---
// Account Holder: rohan
// Account Number: 485485485485
// Current Balance: $58000.0

// Bank Operations Menu:
// 1. Deposit
// 2. Withdraw
// 3. Check Balance
// 4. Exit
// Enter your choice: 4
// Exiting the program.
```

slip7_2.java

```
//? Write a program to accept a text file from user and display the contents of a file in
//? reverse order and change its case.


import java.io.*;
import java.util.Scanner;

public class slip7_2 {
    public static void main(String[] args) {
        String path = "sample.txt"; // Change this to your file's name

        try {
            File file = new File(path);
            Scanner fileSc = new Scanner(file);
            StringBuilder content = new StringBuilder();

            while (fileSc.hasNextLine()) {
                content.append(fileSc.nextLine()).append("\n");
            }
            fileSc.close();

            String result = revCase(content.toString());
            System.out.println("Modified Contents:\n" + result);

        } catch (FileNotFoundException e) {
            System.out.println("File not found: " + e.getMessage());
        }
    }

    // Reverse and change case
    private static String revCase(String input) {
        StringBuilder rev = new StringBuilder();
        for (int i = input.length() - 1; i >= 0; i--) {
            char c = input.charAt(i);
            rev.append(Character.isLowerCase(c) ? Character.toUpperCase(c) :
                        Character.isUpperCase(c) ? Character.toLowerCase(c) : c);
        }
        return rev.toString();
    }
}

// output =
// MAr EERHs IAj , GNINROm DOOg OLLEh
```

sample.txt

Hello Good Morning , Jai Shree Ram

slip8_1.java

```java
// Create a class Sphere, to calculate the volume and surface area of sphere.
// (Hint : Surface area=4*3.14(r*r), Volume=(4/3)3.14(r*r*r))

import java.util.Scanner;

class Sphere {
    private double radius;

    // Constructor to initialize the radius
    public Sphere(double radius) {
        this.radius = radius;
    }

    // Method to calculate surface area
    public double surfaceArea() {
        return 4 * Math.PI * (radius * radius);
    }

    // Method to calculate volume
    public double volume() {
        return (4.0 / 3.0) * Math.PI * (radius * radius * radius);
    }
}

public class slip8_1 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        // Accept radius input from user
        System.out.print("Enter the radius of the sphere: ");
        double radius = sc.nextDouble();

        // Create a Sphere object
        Sphere sphere = new Sphere(radius);

        // Calculate and display surface area and volume
        System.out.printf("Surface Area: %.2f%n", sphere.surfaceArea());
        System.out.printf("Volume: %.2f%n", sphere.volume());

        sc.close();
    }
}

// output =
// Enter the radius of the sphere: 2.5
// Surface Area: 78.54
// Volume: 65.45
```

slip8_2.java

```java
//?  Design a screen to handle the Mouse Events such as MOUSE_MOVED
//?  and MOUSE_CLICKED and display the position of the Mouse_Click in a TextField.

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


public class slip8_2 {
    public static void main(String[] args) {

        JFrame frame = new JFrame("Mouse Event Example");
        frame.setSize(400, 300);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setLayout(null);

        JTextField textField = new JTextField();
        textField.setBounds(50, 200, 300, 50);
        frame.add(textField);

        frame.addMouseListener(new MouseAdapter() {
            @Override
            public void mouseClicked(MouseEvent e) {
                // Display mouse click coordinates in the text field
                textField.setText("Mouse Clicked at: (" + e.getX() + ", " + e.getY() + ")");
            }
        });

        frame.addMouseMotionListener(new MouseAdapter() {
            @Override
            public void mouseMoved(MouseEvent e) {

                // System.out.println("Mouse Moved at: (" + e.getX() + ", " + e.getY() + ")");
            }
        });

        // Make the frame visible
        frame.setVisible(true);
    }
}
```

slip9_1.java

```
//?   Define a “Clock™ class that does the following ;
//?   a. Accept Hours, Minutes and Seconds
//?   b. Check the validity of numbers
//?   c. Set the time to AM/PM mode
//?   Use the necessary constructors and methods to do the above task


import java.util.Scanner;

class Clock {
    private int hours;
    private int minutes;
    private int seconds;
    private String period; // AM or PM

    // Constructor to initialize time
    public Clock(int hours, int minutes, int seconds, String period) {
        if (isValidTime(hours, minutes, seconds, period)) {
            this.hours = hours;
            this.minutes = minutes;
            this.seconds = seconds;
            this.period = period.toUpperCase(); // Ensure AM/PM is in uppercase
        } else {
            throw new IllegalArgumentException("Invalid time provided.");
        }
    }

    // Method to validate time
    private boolean isValidTime(int hours, int minutes, int seconds, String period) {
        if (period.equalsIgnoreCase("AM") || period.equalsIgnoreCase("PM")) {
            if (hours < 1 || hours > 12) return false; // Valid hours: 1 to 12
            if (minutes < 0 || minutes > 59) return false; // Valid minutes: 0 to 59
            if (seconds < 0 || seconds > 59) return false; // Valid seconds: 0 to 59
            return true;
        }
        return false;
    }

    // Method to display the time in HH:MM:SS AM/PM format
    public void displayTime() {
        System.out.printf("Time: %02d:%02d:%02d %s%n", hours, minutes, seconds, period);
    }

    // Method to set the time in AM/PM mode
    public void setTime(int hours, int minutes, int seconds, String period) {
        if (isValidTime(hours, minutes, seconds, period)) {
            this.hours = hours;
            this.minutes = minutes;
            this.seconds = seconds;
            this.period = period.toUpperCase();
        } else {
```

```
            throw new IllegalArgumentException("Invalid time provided.");
        }
    }
}

public class slip9_1 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        // Accept time input from the user
        System.out.print("Enter hours (1-12): ");
        int hours = sc.nextInt();
        System.out.print("Enter minutes (0-59): ");
        int minutes = sc.nextInt();
        System.out.print("Enter seconds (0-59): ");
        int seconds = sc.nextInt();
        sc.nextLine(); // Consume newline
        System.out.print("Set the period of time (AM/PM) : ");
        String period = sc.nextLine();

        try {
            // Create a Clock object
            Clock clock = new Clock(hours, minutes, seconds, period);
            clock.displayTime(); // Display the set time

        } catch (IllegalArgumentException e) {
            System.out.println(e.getMessage());
        }
        sc.close();
    }
}

// output =
// Enter hours (1-12): 2
// Enter minutes (0-59): 52
// Enter seconds (0-59): 54
// Set the period of time (AM/PM) : pM
// Time: 02:52:54 PM
```

slip9_2.java

```java
//?   Write a program to using marker interface create a class Product
//?   (product_id, product_name, product_cost, product_quantity) default and
//?   parameterized constructor. Create objectsof class  product and
//?   display the contents of each object and Also display the object count.

interface PMarker {
}

class P implements PMarker {
    private static int cnt = 0;
    private int id;
    private String name;
    private double cost;
    private int qty;

    public P() {
        this.id = 0;
        this.name = "Unknown";
        this.cost = 0.0;
        this.qty = 0;
        cnt++;
    }

    public P(int id, String name, double cost, int qty) {
        this.id = id;
        this.name = name;
        this.cost = cost;
        this.qty = qty;
        cnt++;
    }

    public void show() {
        System.out.printf("ID: %d%n", id);
        System.out.printf("Name: %s%n", name);
        System.out.printf("Cost: %.2f%n", cost);
        System.out.printf("Qty: %d%n", qty);
        System.out.println("-----------------------------");
    }

    public static int getCnt() {
        return cnt;
    }
}

public class slip9_2 {
    public static void main(String[] args) {
        P p1 = new P(101, "Laptop", 75000.00, 10);
        P p2 = new P(102, "Smartphone", 30000.00, 25);
        P p3 = new P();

        System.out.println("Product Details:");
```

```
        p1.show();
        p2.show();
        p3.show();

        System.out.println("Total Products: " + P.getCnt());
    }
}

// output =
// Product Details:
// ID: 101
// Name: Laptop
// Cost: 75000.00
// Qty: 10
// ----------------------------
// ID: 102
// Name: Smartphone
// Cost: 30000.00
// Qty: 25
// ----------------------------
// ID: 0
// Name: Unknown
// Cost: 0.00
// Qty: 0
// ----------------------------
// Total Products: 3
```

slip10_1.java

```java
//?  Write a program to find the cube of given number using
//?   functional interface.

import java.util.Scanner;

interface Cube {
    int calculate(int n);
}

public class slip10_1 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter a number: ");
        int number = scanner.nextInt();

        Cube cube = (n) -> n * n * n;
        int result = cube.calculate(number);

        System.out.println("Cube of " + number + " is: " + result);

        scanner.close();
    }
}


// output =
// Enter a number: 2
// Cube of 2 is: 8
```

StudentInfo.java

slip10_2

```
package student;

public class StudentInfo {
    private int rollNo;
    private String name;
    private String className;
    private double percentage;

    // Constructor
    public StudentInfo(int rollNo, String name, String
className, double percentage) {
        this.rollNo = rollNo;
        this.name = name;
        this.className = className;
        this.percentage = percentage;
    }

    // Method to display student information
    public void displayInfo() {
        System.out.println("Student Roll No: " + rollNo);
        System.out.println("Student Name: " + name);
        System.out.println("Class: " + className);
        System.out.println("Percentage: " + percentage + "%");
    }
}
```

StudentPer.java

```
//?   Write a program to create a package name student. Define
class StudentInfo with method to
//?   display information about student such as rollno, class,
and percentage. Create another class
//?   StudentPer with method to find percentage of the student.
Accept student details like
//?   rollno, name, class and marks of 6 subject from user.
//?   package student;
package student;
import java.util.Scanner;

public class StudentPer {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        // Accept student details
        System.out.print("Enter Roll No: ");
        int rollNo = sc.nextInt();
        sc.nextLine(); // Consume newline

        System.out.print("Enter Name: ");
        String name = sc.nextLine();

        System.out.print("Enter Class: ");
        String className = sc.nextLine();

        int[] marks = new int[6];
        int totalMarks = 0;

        // Accept marks for 6 subjects
        for (int i = 0; i < 6; i++) {
            System.out.print("Enter marks for subject " + (i +
1) + ": ");
            marks[i] = sc.nextInt();
            totalMarks += marks[i];
```

```
        }

        // Calculate percentage
        double percentage = (double) totalMarks / 6;

        // Create an object of StudentInfo and display
information
        StudentInfo student = new StudentInfo(rollNo, name,
className, percentage);
        student.displayInfo();

        sc.close();
    }
}

//*     open terminal run follwing commands =
//*     javac -d . *.java (all java files in this folder create
package )
//*     java StudentPer.java (for running this programe)

// output =
// Enter Roll No: 45
// Enter Name: rohit
// Enter Class: fy
// Enter marks for subject 1: 85
// Enter marks for subject 2: 85
// Enter marks for subject 3: 95
// Enter marks for subject 4: 95
// Enter marks for subject 5: 80
// Enter marks for subject 6: 85
// Student Roll No: 45
// Student Name: rohit
// Class: fy
// Percentage: 87.5%
```

slip11_1Cylinder.java

```
// Define an interface “Operation” which has method volume( ).
// Define a constant PI having a value 3.142 Create a class cylinder
//  which implements this interface (members – radius,height).
// Create one object and calculate the volume.
import java.util.Scanner;

interface Operation {
    double PI = 3.142;
    double volume();
}

public class slip11_1Cylinder implements Operation {
    private double radius;
    private double height;

    public slip11_1Cylinder(double radius, double height) {
        this.radius = radius;
        this.height = height;
    }

    @Override
    public double volume() {
        return PI * radius * radius * height;
    }

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.print("Enter radius of the cylinder: ");
        double radius = sc.nextDouble();
        System.out.print("Enter height of the cylinder: ");
        double height = sc.nextDouble();

        slip11_1Cylinder cylinder = new slip11_1Cylinder(radius, height);

        double volume = cylinder.volume();
        System.out.println("Volume of the cylinder: " + volume);

        sc.close();
    }
}

// output =
// Enter radius of the cylinder: 2
// Enter height of the cylinder: 2
// Volume of the cylinder: 25.136
```

slip11_2UserAuth.java

```
// Write a program to accept the username and password from user
// if username and password are not same then raise
//  "Invalid Password" with appropriate msg.


import java.util.Scanner;

class InvalidPwdException extends Exception {
    public InvalidPwdException(String msg) {
        super(msg);
    }
}

public class slip11_2UserAuth {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        try {
            // Accept username and password from user
            System.out.print("Enter Username: ");
            String user = sc.nextLine();

            System.out.print("Enter Password: ");
            String pwd = sc.nextLine();

            // Check if username and password are the same
            if (!user.equals(pwd)) {
                throw new InvalidPwdException("Invalid Password: Username and password must
match.");
            }

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

        } catch (InvalidPwdException e) {
            System.out.println(e.getMessage());
        } finally {
            sc.close();
        }
    }
}

// output =
// Enter Username: rohit@999
// Enter Password: rohit@999
// Login successful!
```

slip14_1.java

```
//? Write a program to accept a number from the user,
//? if number is zero then throw user defined exception
//? “Number is 0” otherwise check whether
//? number is prime or not (Use static keyword).


import java.util.Scanner;

class ZeroEx extends Exception {
    public ZeroEx(String msg) {
        super(msg);
    }
}

public class slip14_1 {
    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;
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        try {
            System.out.print("Enter a number: ");
            int n = sc.nextInt();

            if (n == 0) {
                throw new ZeroEx("Number is 0");
            }

            if (isPrime(n)) {
                System.out.println(n + " is a prime number.");
            } else {
                System.out.println(n + " is not a prime number.");
            }

        } catch (ZeroEx e) {
            System.out.println(e.getMessage());
        } finally {
            sc.close();
        }
    }
}

// output =
// Enter a number: 5
// 5 is a prime number.
```

slip14_2

SY\SM.java

```
package SY;

public class SM {
    public int cTotal;
    public int mTotal;
    public int eTotal;

    public SM(int cTotal, int mTotal, int eTotal) {
        this.cTotal = cTotal;
        this.mTotal = mTotal;
        this.eTotal = eTotal;
    }
}
```

TY\TM.java

```java
package TY;

public class TM {
    public int theory;
    public int practicals;

    public TM(int theory, int practicals) {
        this.theory = theory;
        this.practicals = practicals;
    }
}
```

student.java

```
//?  Write a Java program to create a Package “SY” which has a class SYMarks (members –
//?  ComputerTotal, MathsTotal, and ElectronicsTotal). Create another package TY which has a
//?  class TYMarks (members – Theory, Practicals). Create ‘n” objects of Student class (having
//?  rolINumber, name, SYMarks and TYMarks). Add the marks of SY and TY computer subjects
//?  and calculate the Grade (‘A” for >= 70, ‘B’ for >= 60 ‘C” for >= 50, Pass Class for > =40
//?  else‘FAIL’) and display the result of the student in proper format.


import java.util.Scanner;
import SY.SM;
import TY.TM;

public class student {
    private int roll;
    private String name;
    private SM sm;
    private TM tm;

    public student(int roll, String name, SM sm, TM tm) {
        this.roll = roll;
        this.name = name;
        this.sm = sm;
        this.tm = tm;
    }

    public char calcGrade() {
        int total = sm.cTotal + tm.theory + tm.practicals;
        if (total >= 70) return 'A';
        else if (total >= 60) return 'B';
        else if (total >= 50) return 'C';
        else if (total >= 40) return 'P';
        else return 'F';
    }

    public void showResult() {
        char grade = calcGrade();
        System.out.println("Roll No: " + roll);
        System.out.println("Name: " + name);
        System.out.println("Grade: " + grade);
    }

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

        for (int i = 0; i < n; i++) {
            System.out.print("Enter Roll No for Student " + (i + 1) + ": ");
            int roll = sc.nextInt();
            sc.nextLine(); // Consume newline
```

```
            System.out.print("Enter Name for Student " + (i + 1) + ": ");
            String name = sc.nextLine();

            System.out.print("Enter SY Computer Total: ");
            int cTotal = sc.nextInt();
            System.out.print("Enter SY Maths Total: ");
            int mTotal = sc.nextInt();
            System.out.print("Enter SY Electronics Total: ");
            int eTotal = sc.nextInt();
            SM sm = new SM(cTotal, mTotal, eTotal);

            System.out.print("Enter TY Theory Marks: ");
            int theory = sc.nextInt();
            System.out.print("Enter TY Practical Marks: ");
            int practicals = sc.nextInt();
            TM tm = new TM(theory, practicals);

            students[i] = new student(roll, name, sm, tm);
        }

        System.out.println("\nResults:");
        for (student student : students) {
            student.showResult();
        }

        sc.close();
    }
}

//*   javac -d . *.java
//*   java student.java

// output =
// Enter number of students: 3
// Enter Roll No for Student 1: 11
// Enter Name for Student 1: suresh
// Enter SY Computer Total: 400
// Enter SY Maths Total: 300
// Enter SY Electronics Total: 350
// Enter TY Theory Marks: 400
// Enter TY Practical Marks: 100
// Enter Roll No for Student 2: 12
// Enter Name for Student 2: ritesh
// Enter SY Computer Total: 400
// Enter SY Maths Total: 456
// Enter SY Electronics Total: 125
// Enter TY Theory Marks: 500
// Enter TY Practical Marks: 100
// Enter Roll No for Student 3: 13
// Enter Name for Student 3: mayur
// Enter SY Computer Total: 300
// Enter SY Maths Total: 350
// Enter SY Electronics Total: 400
```

```
// Enter TY Theory Marks: 120
// Enter TY Practical Marks: 120

// Results:
// Roll No: 11
// Name: suresh
// Grade: A
// Roll No: 12
// Name: ritesh
// Grade: A
// Roll No: 13
// Name: mayur
// Grade: A
```

slip15_1.java

```
// Accept the names of two files and copy the contents of the
// first to the second. First file having Book name and
// Author name in file.

import java.io.*;
import java.util.Scanner;

public class slip15_1 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Enter source file name (with .txt): ");
        String srcFile = sc.nextLine();

        System.out.print("Enter destination file name (with .txt): ");
        String destFile = sc.nextLine();

        File src = new File(srcFile);
        File dest = new File(destFile);

        try (FileReader fr = new FileReader(src); FileWriter fw = new FileWriter(dest)) {
            int ch;
            while ((ch = fr.read()) != -1) {
                fw.write(ch);
            }
            System.out.println("Contents copied from " + srcFile + " to " + destFile);
        } catch (IOException e) {
            System.out.println("Error: " + e.getMessage());
        } finally {
            sc.close();
        }
    }
}


// output =
// Enter source file name (with .txt): slip15f1.txt
// Enter destination file name (with .txt): slip15f2.txt
// Contents copied from slip15f1.txt to slip15f2.txt
```

slip15_2.java

```
// Write a program to define a class Account having members custname, accno. Define default
// and parameterized constructor. Create a subclass called SavingAccount with member savingbal,
// minbal. Create a derived class AccountDetail that extends the class SavingAccount with
// members, depositamt and withdrawalamt. Write a appropriate method to display customer
// details.


class Acc {
    String cName;
    String accNo;

    Acc() {
        cName = "Unknown";
        accNo = "0000";
    }

    Acc(String cName, String accNo) {
        this.cName = cName;
        this.accNo = accNo;
    }
}

class SavAcc extends Acc {
    double sBal;
    double minBal;

    SavAcc() {
        super();
        sBal = 0.0;
        minBal = 1000.0;
    }

    SavAcc(String cName, String accNo, double sBal, double minBal) {
        super(cName, accNo);
        this.sBal = sBal;
        this.minBal = minBal;
    }
}

class AccDetail extends SavAcc {
    double depAmt;
    double withAmt;

    AccDetail() {
        super();
        depAmt = 0.0;
        withAmt = 0.0;
    }

    AccDetail(String cName, String accNo, double sBal, double minBal, double depAmt, double
withAmt) {
        super(cName, accNo, sBal, minBal);
```

```
        this.depAmt = depAmt;
        this.withAmt = withAmt;
    }

    void showDetails() {
        System.out.println("Customer Name: " + cName);
        System.out.println("Account Number: " + accNo);
        System.out.println("Saving Balance: " + sBal);
        System.out.println("Minimum Balance: " + minBal);
        System.out.println("Deposit Amount: " + depAmt);
        System.out.println("Withdrawal Amount: " + withAmt);
    }
}

public class slip15_2 {
    public static void main(String[] args) {
        AccDetail accDetail = new AccDetail("John Doe", "12345", 5000.0, 1000.0, 1500.0, 200.0);
        accDetail.showDetails();
    }
}

// output =
// Customer Name: John Doe
// Account Number: 12345
// Saving Balance: 5000.0
// Minimum Balance: 1000.0
// Deposit Amount: 1500.0
// Withdrawal Amount: 200.0
```

slip15f1.txt

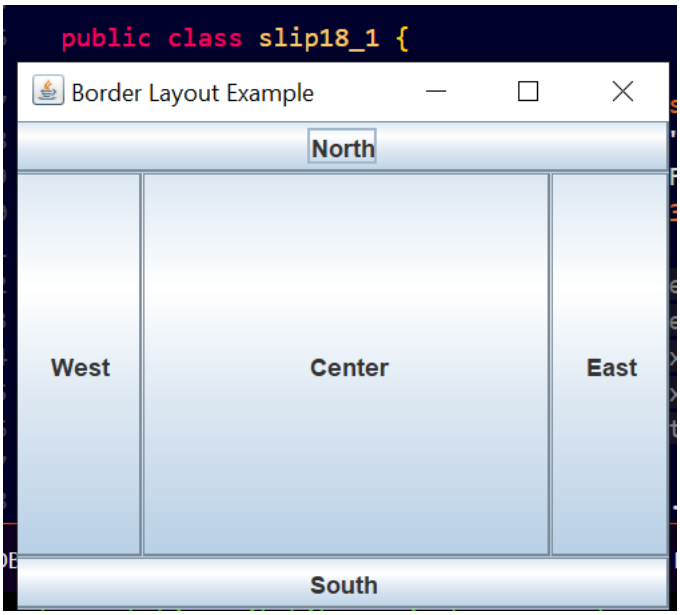
```
"The Catcher in the Rye"  ================>     by J.D. Salinger
"To Kill a Mockingbird"   ================>     by Harper Lee
"1984"                    ================>     by George Orwell
"Pride and Prejudice"     ================>     by Jane Austen
"The Great Gatsby"        ================>     by F. Scott Fitzgerald
"Moby-Dick"               ================>     by Herman Melville
"War and Peace"           ================>     by Leo Tolstoy
"Brave New World"         ================>     by Aldous Huxley
"The Hobbit"              ================>     by J.R.R. Tolkien
"Fahrenheit 451"          ================>     by Ray Bradbury
```

slip15f2.txt

```
 Gatsby"        ================>     by F. Scott Fitzgerald
"Moby-Dick"               ================>     by Herman Melville
"War and Peace"           ================>     by Leo Tolstoy
"Brave New World"         ================>     by Aldous Huxley
"The Hobbit"              ================>     by J.R.R. Tolkien
"Fahrenheit 451"          ================>     by Ray Bradbury
```

slip18_1.java

```java
// Write a program to implement Border Layout Manager.

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

public class slip18_1 {
    public static void main(String[] args) {
        JFrame frame = new JFrame("Border Layout Example");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(400, 300);

        JButton btnNorth = new JButton("North");
        JButton btnSouth = new JButton("South");
        JButton btnEast = new JButton("East");
        JButton btnWest = new JButton("West");
        JButton btnCenter = new JButton("Center");

        frame.add(btnNorth, BorderLayout.NORTH);
        frame.add(btnSouth, BorderLayout.SOUTH);
        frame.add(btnEast, BorderLayout.EAST);
        frame.add(btnWest, BorderLayout.WEST);
        frame.add(btnCenter, BorderLayout.CENTER);

        frame.setVisible(true);
    }
}
```



slip18_2.java

```
//?   Define a class CricketPlayer (name,no_of_innings,no_of_times_notout, totatruns, bat_avg).
//?   Create an array of n player objects. Calculate the batting average for each player using
static
//?   method avg(). Define a static sort method which sorts the array on the basis of average.
//?   Display the player details in sorted order.

import java.util.*;

class Player {
    String n;
    int innings;
    int notOut;
    int runs;
    double avg;

    Player(String n, int innings, int notOut, int runs) {
        this.n = n;
        this.innings = innings;
        this.notOut = notOut;
        this.runs = runs;
        this.avg = calcAvg(runs, innings, notOut);
    }

    static double calcAvg(int runs, int innings, int notOut) {
        return innings > 0 ? (double) runs / (innings - notOut) : 0.0;
    }

    static void sort(Player[] p) {
        Arrays.sort(p, Comparator.comparingDouble(player -> player.avg));
    }

    void display() {
        System.out.printf("Name: %s, Innings: %d, Not Out: %d, Total Runs: %d, Batting Average:
%.2f%n",
                n, innings, notOut, runs, avg);
    }
}

public class slip18_2 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Enter number of players: ");
        int n = sc.nextInt();
        Player[] p = new Player[n];

        for (int i = 0; i < n; i++) {
            System.out.print("Enter name of player " + (i + 1) + ": ");
            String name = sc.next();
            System.out.print("Enter number of innings: ");
            int innings = sc.nextInt();
            System.out.print("Enter number of times not out: ");
```

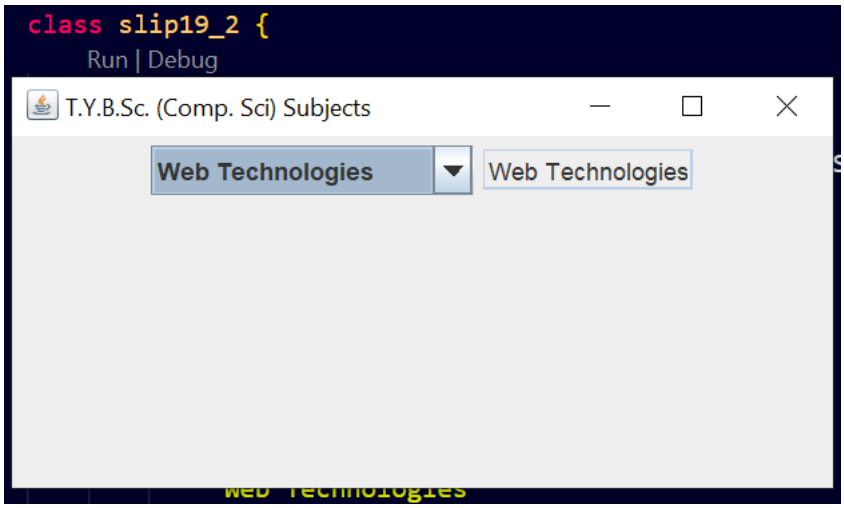
```
            int notOut = sc.nextInt();
            System.out.print("Enter total runs: ");
            int runs = sc.nextInt();

            p[i] = new Player(name, innings, notOut, runs);
        }

        Player.sort(p);

        System.out.println("\nPlayer details in sorted order by batting average:");
        for (Player player : p) {
            player.display();
        }

        sc.close();
    }
}

// output =
// Enter number of players: 2
// Enter name of player 1: rohit
// Enter number of innings: 2
// Enter number of times not out: 1
// Enter total runs: 100
// Enter name of player 2: xyz
// Enter number of innings: 3
// Enter number of times not out: 1
// Enter total runs: 99

// Player details in sorted order by batting average:
// Name: xyz, Innings: 3, Not Out: 1, Total Runs: 99, Batting Average: 49.50
// Name: rohit, Innings: 2, Not Out: 1, Total Runs: 100, Batting Average: 100.00
```

slip19_1.java

```java
import java.util.Scanner;

public class slip19_1 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Enter the number of rows: ");
        int rows = sc.nextInt();
        System.out.print("Enter the number of columns: ");
        int cols = sc.nextInt();

        int[][] matrix = new int[rows][cols];

        System.out.println("Enter the elements of the matrix:");
        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < cols; j++) {
                matrix[i][j] = sc.nextInt();
            }
        }

        int primaryDiagonalSum = 0;
        int secondaryDiagonalSum = 0;

        for (int i = 0; i < rows; i++) {
            primaryDiagonalSum += matrix[i][i]; // Sum for primary diagonal
            secondaryDiagonalSum += matrix[i][cols - 1 - i]; // Sum for secondary diagonal
        }

        System.out.println("Sum of primary diagonal: " + primaryDiagonalSum);
        System.out.println("Sum of secondary diagonal: " + secondaryDiagonalSum);

        sc.close();
    }
}

// output =
// Enter the number of rows: 3
// Enter the number of columns: 3
// Enter the elements of the matrix:
// 1
// 2
// 3
// 4
// 5
// 6
// 7
// 8
// 9
// Sum of primary diagonal: 15
// Sum of secondary diagonal: 15
```

slip19_2.java

```java
// Write a program which shows the combo box which includes list of T.Y.B.Sc.(Comp. Sci)
// subjects. Display the selected subject in a text field.

import javax.swing.*;

class slip19_2 {
    public static void main(String[] args) {
        JFrame f = new JFrame("T.Y.B.Sc. (Comp. Sci) Subjects");
        f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        f.setSize(300, 100);

        String[] s = {
            "Data Structures",
            "Database Management",
            "Operating Systems",
            "Computer Networks",
            "Software Engineering",
            "Web Technologies"
        };

        JComboBox<String> cb = new JComboBox<>(s);
        JTextField tf = new JTextField();
        tf.setEditable(false);

        cb.addActionListener(e -> tf.setText((String) cb.getSelectedItem()));

        JPanel p = new JPanel();
        p.add(cb);
        p.add(tf);

        f.add(p);
        f.setVisible(true);
    }
}
```



slip20_1.java

```
// * this program also in slip5_1

// output =
// continent Name : Asia
// County Name : India
// State Name :Maharashtra
// place name pune
```

slip20_2

Operation\Addition.java

```java
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;
    }
}
```

Operation\Maximum.java

```java
package Operation;

public class Maximum {
    public int max(int a, int b) {
        return (a > b) ? a : b;
    }
}
```

op.java

```
// Write a package for Operation, which has two classes,
Addition and Maximum. Addition has
// two methods add () and subtract (), which are used to add two
integers and subtract two,
// float values respectively. Maximum has a method max () to
display the maximum of two
// integers

import Operation.Addition;
import Operation.Maximum;

public class op {
    public static void main(String[] args) {
        Addition add = new Addition();
        Maximum max = new Maximum();

        // Addition
        int sum = add.add(5, 10);
        float diff = add.subtract(15.5f, 10.2f);

        System.out.println("Sum: " + sum);
        System.out.println("Difference: " + diff);

        // Maximum
        int maximum = max.max(20, 35);
        System.out.println("Maximum: " + maximum);
    }
}

//* javac -d . Addition.java
//* javac -d .Maximum.java
//* java op

// output =
// Sum: 15
```

```
  // Difference: 5.3
  // Maximum: 35
```

slip21_1.java

```
// Define a class MyDate(Day, Month, year) with methods to accept and display a MyDateobject.
// Accept date as dd,mm,yyyy. Throw user defined exception "InvalidDateException” if the date
// is invalid.

import java.util.Scanner;

class InvalidDateException extends Exception {
    public InvalidDateException(String message) {
        super(message);
    }
}

class MyDate {
    private int day;
    private int month;
    private int year;

    public void acceptDate() throws InvalidDateException {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter date (dd mm yyyy): ");
        day = sc.nextInt();
        month = sc.nextInt();
        year = sc.nextInt();
        sc.close();

        if (!isValidDate(day, month, year)) {
            throw new InvalidDateException("Invalid Date: " + day + "/" + month + "/" + year);
        }

    }

    private boolean isValidDate(int d, int m, int y) {
        if (y < 1) return false;
        if (m < 1 || m > 12) return false;

        int[] daysInMonth = {31, (isLeapYear(y) ? 29 : 28), 31, 30, 31, 30, 31, 31, 30, 31, 30,
31};
        return d > 0 && d <= daysInMonth[m - 1];
    }

    private boolean isLeapYear(int y) {
        return (y % 4 == 0 && y % 100 != 0) || (y % 400 == 0);
    }

    public void displayDate() {
        System.out.printf("Date: %02d/%02d/%d%n", day, month, year);
    }
}

public class slip21_1 {
    public static void main(String[] args) {
        MyDate date = new MyDate();
```

```
        try {
            date.acceptDate();
            date.displayDate();
        } catch (InvalidDateException e) {
            System.out.println(e.getMessage());
        }
    }
}

// output =
// Enter date (dd mm yyyy): 12
// 03
// 2020
// Date: 12/03/2020
```

slip21_2.java

```
// Create an employee class(id,name,deptname,salary). Define a default and parameterized
// constructor. Use ‘this” keyword to initialize instance variables. Keep a count of objects
// created. Create objects using parameterized constructor and display the object count after
// each object is created. (Use static member and method). Also display the contents of each
// object.


class Employee {
    private int id;
    private String name;
    private String deptName;
    private double salary;
    private static int count = 0; // Static variable to count objects

    // Default constructor
    public Employee() {
        this.id = 0;
        this.name = "Unknown";
        this.deptName = "Not Assigned";
        this.salary = 0.0;
        count++;
        display();
    }

    // Parameterized constructor
    public Employee(int id, String name, String deptName, double salary) {
        this.id = id;
        this.name = name;
        this.deptName = deptName;
        this.salary = salary;
        count++;
        display();
    }

    // Static method to get the object count
    public static int getObjectCount() {
        return count;
    }

    // Method to display employee details
    public void display() {
        System.out.println("Employee ID: " + id);
        System.out.println("Name: " + name);
        System.out.println("Department: " + deptName);
        System.out.println("Salary: " + String.format("%.2f", salary));
        System.out.println("Total Employees Created: " + getObjectCount());
    }
}

public class slip21_2 {
    public static void main(String[] args) {
```

```
        Employee emp1 = new Employee(101, "Alice", "IT", 60000);
        Employee emp2 = new Employee(102, "Bob", "HR", 50000);
        Employee emp3 = new Employee(103, "Charlie", "Finance", 70000);
        Employee emp4 = new Employee();

        System.out.println("Final Count of Employees: " + Employee.getObjectCount());
    }
}

// ouitput =
// Employee ID: 101
// Name: Alice
// Department: IT
// Salary: 60000.00
// Total Employees Created: 1
// Employee ID: 102
// Name: Bob
// Department: HR
// Salary: 50000.00
// Total Employees Created: 2
// Employee ID: 103
// Name: Charlie
// Department: Finance
// Salary: 70000.00
// Total Employees Created: 3
// Employee ID: 0
// Name: Unknown
// Department: Not Assigned
// Salary: 0.00
// Total Employees Created: 4
// Final Count of Employees: 4
```

slip25_1.java

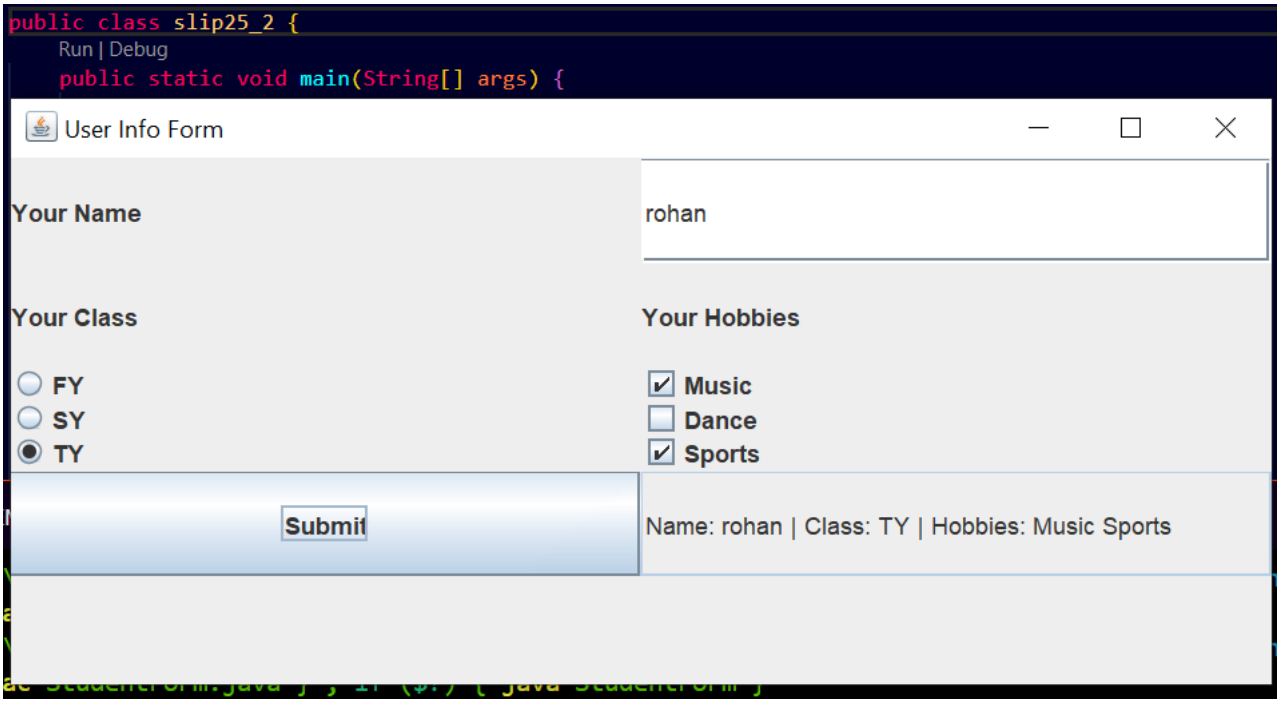
```
// Create a class Student(rollno, name ,class, per), to read student information from the
console
// and display them (Using BufferedReader class)


import java.io.*;

class Student {
    int rollNo;
    String name, cls; // Changed to 'cls' to avoid conflict with the keyword 'class'
    float per;

    // Constructor to initialize Student attributes
    public Student(int rollNo, String name, String cls, float per) {
        this.rollNo = rollNo;
        this.name = name;
        this.cls = cls;
        this.per = per;
    }

    // Method to display student information
    public void display() {
        System.out.printf("Roll No: %d, Name: %s, Class: %s, Percentage: %.2f%%%n", rollNo,
name, cls, per);
    }
}

public class slip25_1 {
    public static void main(String[] args) {
        try (BufferedReader reader = new BufferedReader(new InputStreamReader(System.in))) {
            Student[] students = new Student[5]; // Array to store 5 Student objects

            // Read information for 5 Student objects
            for (int i = 0; i < 5; i++) {
                System.out.print("Enter Roll No: ");
                int rollNo = Integer.parseInt(reader.readLine());
                System.out.print("Enter Name: ");
                String name = reader.readLine();
                System.out.print("Enter Class: ");
                String cls = reader.readLine();
                System.out.print("Enter Percentage: ");
                float per = Float.parseFloat(reader.readLine());
                students[i] = new Student(rollNo, name, cls, per); // Create a new Student
object
            }

            System.out.println("\nStudent Information:");
            for (Student student : students) {
                student.display(); // Display each student's information
            }
        } catch (IOException | NumberFormatException e) {
```

```
            System.out.println("Error: " + e.getMessage());
        }
    }
}



// output =
// Enter Roll No: 101
// Enter Name: rohit
// Enter Class: fy
// Enter Percentage: 85
// Enter Roll No: 102
// Enter Name: mayur
// Enter Class: fy
// Enter Percentage: 86
// Enter Roll No: 103
// Enter Name: puja
// Enter Class: ty
// Enter Percentage: 96
// Enter Roll No: 104
// Enter Name: pornema
// Enter Class: sy
// Enter Percentage: 90
// Enter Roll No: 105
// Enter Name: netal
// Enter Class: fy
// Enter Percentage: 85

// Student Information:
// Roll No: 101, Name: rohit, Class: fy, Percentage: 85.00%
// Roll No: 102, Name: mayur, Class: fy, Percentage: 86.00%
// Roll No: 103, Name: puja, Class: ty, Percentage: 96.00%
// Roll No: 104, Name: pornema, Class: sy, Percentage: 90.00%
// Roll No: 105, Name: netal, Class: fy, Percentage: 85.00%
```

slip25_2.java

```
// Create the following GUI screen using appropriate layout manager. Accept the name, class,
// hobbies from the user and display the selected options in a textbox.
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class slip25_2 {
    public static void main(String[] args) {
        // Frame creation
        JFrame frame = new JFrame("User Info Form");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(400, 300);

        // Create a panel with GridLayout
        JPanel panel = new JPanel();
        panel.setLayout(new GridLayout(5, 2)); // Adjusting the rows to align correctly

        // Your Name
        JLabel nameLabel = new JLabel("Your Name");
        JTextField nameField = new JTextField();
        panel.add(nameLabel);
        panel.add(nameField);

        // Class label and hobbies label
        JLabel classLabel = new JLabel("Your Class");
        JLabel hobbiesLabel = new JLabel("Your Hobbies");
        panel.add(classLabel);
        panel.add(hobbiesLabel);

        // Class selection with radio buttons in the left column
        JPanel classPanel = new JPanel(new GridLayout(3, 1));
        JRadioButton fyRadio = new JRadioButton("FY");
        JRadioButton syRadio = new JRadioButton("SY");
        JRadioButton tyRadio = new JRadioButton("TY");
        ButtonGroup classGroup = new ButtonGroup();
        classGroup.add(fyRadio);
        classGroup.add(syRadio);
        classGroup.add(tyRadio);
        classPanel.add(fyRadio);
        classPanel.add(syRadio);
        classPanel.add(tyRadio);
        panel.add(classPanel);

        // Hobbies selection with checkboxes in the right column
        JPanel hobbiesPanel = new JPanel(new GridLayout(3, 1));
        JCheckBox musicCheckBox = new JCheckBox("Music");
        JCheckBox danceCheckBox = new JCheckBox("Dance");
        JCheckBox sportsCheckBox = new JCheckBox("Sports");
        hobbiesPanel.add(musicCheckBox);
        hobbiesPanel.add(danceCheckBox);
```

```
        hobbiesPanel.add(sportsCheckBox);
        panel.add(hobbiesPanel);

        // Output TextField
        JTextField outputField = new JTextField();
        outputField.setEditable(false);

        // Submit Button
        JButton submitButton = new JButton("Submit");
        submitButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                String name = nameField.getText();
                String selectedClass = fyRadio.isSelected() ? "FY" :
                        syRadio.isSelected() ? "SY" : tyRadio.isSelected() ? "TY" : "";
                String hobbies = "";
                if (musicCheckBox.isSelected()) hobbies += "Music ";
                if (danceCheckBox.isSelected()) hobbies += "Dance ";
                if (sportsCheckBox.isSelected()) hobbies += "Sports ";
                outputField.setText("Name: " + name + " | Class: " + selectedClass + " |
Hobbies: " + hobbies);
            }
        });

        // Add components for submission and output display
        panel.add(submitButton);
        panel.add(outputField);

        // Add panel to frame and display
        frame.add(panel);
        frame.setVisible(true);
    }
}
```



slip26_1.java

```
// ? Define a Item class (item_number, item_name, item_price). Define a default and
parameterized
// ? constructor. Keep a count of objects created. Create objects using parameterized
constructor
// ? and display the object count after each object is created.(Use static member and method).
Also
// ? display the contents of each object.

class Item {
    private int itemNumber;
    private String itemName;
    private double itemPrice;
    private static int count = 0; // Static variable to count objects

    // Default constructor
    public Item() {
        this.itemNumber = 0;
        this.itemName = "Unknown";
        this.itemPrice = 0.0;
        count++;
        display();
    }

    // Parameterized constructor
    public Item(int itemNumber, String itemName, double itemPrice) {
        this.itemNumber = itemNumber;
        this.itemName = itemName;
        this.itemPrice = itemPrice;
        count++;
        display();
    }

    // Static method to get the object count
    public static int getObjectCount() {
        return count;
    }

    // Method to display item details
    public void display() {
        System.out.println("Item Number: " + itemNumber);
        System.out.println("Item Name: " + itemName);
        System.out.println("Item Price: " + String.format("%.2f", itemPrice));
        System.out.println("Total Items Created: " + getObjectCount());
        System.out.println();
    }
}

public class slip26_1 {
    public static void main(String[] args) {
        // Creating item objects using the parameterized constructor
        Item item1 = new Item(101, "Laptop", 75000);
        Item item2 = new Item(102, "Mobile", 20000);
```

```
        Item item3 = new Item(103, "Tablet", 15000);

        // Creating an item object using the default constructor
        Item item4 = new Item();

        // Displaying the final count of items created
        System.out.println("Final Count of Items: " + Item.getObjectCount());
    }
}

// output =
// Item Number: 101
// Item Name: Laptop
// Item Price: 75000.00
// Total Items Created: 1

// Item Number: 102
// Item Name: Mobile
// Item Price: 20000.00
// Total Items Created: 2


// Item Number: 103
// Item Name: Tablet
// Item Price: 15000.00
// Total Items Created: 3

// Item Number: 0
// Item Name: Unknown
// Item Price: 0.00
// Total Items Created: 4

// Final Count of Items: 4
```

slip26_2.java

```
// Define a class “Donor’ to store the below mentioned details of a blood donor. name, age,
// address, contactnumber, bloodgroup, date of last donation. Create ‘n’ objects of this class
for
// all the regular donors at Pune. Write these objects to a file. Read these objects from the
file and
// display only those donors’ details whose blood group is “‘A+ve’ and had not donated for the
// recent six months.
import java.io.*;
import java.time.LocalDate;
import java.util.*;


// Donor Class
class Donor implements Serializable {
    private String name;
    private int age;
    private String address;
    private String contactNumber;
    private String bloodGroup;
    private LocalDate lastDonationDate;

    public Donor(String name, int age, String address, String contactNumber, String bloodGroup,
LocalDate lastDonationDate) {
        this.name = name;
        this.age = age;
        this.address = address;
        this.contactNumber = contactNumber;
        this.bloodGroup = bloodGroup;
        this.lastDonationDate = lastDonationDate;
    }

    public String getBloodGroup() {
        return bloodGroup;
    }

    public LocalDate getLastDonationDate() {
        return lastDonationDate;
    }

    @Override
    public String toString() {
        return "Name: " + name + ", Age: " + age + ", Address: " + address + ", Contact: " +
contactNumber +
               ", Blood Group: " + bloodGroup + ", Last Donation Date: " + lastDonationDate;
    }
}

// Main Class for Donor Management
public class slip26_2 {
    private static final String FILE_NAME = "donors.dat";

    @SuppressWarnings("unchecked")
```

```
    public static void main(String[] args) {
        List<Donor> donors = new ArrayList<>();
        Scanner scanner = new Scanner(System.in);

        // Input donor details
        System.out.print("Enter number of donors: ");
        int n = scanner.nextInt();
        scanner.nextLine(); // Consume newline

        for (int i = 0; i < n; i++) {
            System.out.println("Enter details for donor " + (i + 1) + ":");
            System.out.print("Name: ");
            String name = scanner.nextLine();
            System.out.print("Age: ");
            int age = scanner.nextInt();
            scanner.nextLine(); // Consume newline
            System.out.print("Address: ");
            String address = scanner.nextLine();
            System.out.print("Contact Number: ");
            String contactNumber = scanner.nextLine();
            System.out.print("Blood Group: ");
            String bloodGroup = scanner.nextLine();
            System.out.print("Last Donation Date (yyyy-mm-dd): ");
            LocalDate lastDonationDate = LocalDate.parse(scanner.nextLine());

            donors.add(new Donor(name, age, address, contactNumber, bloodGroup,
lastDonationDate));
        }

        // Write donors to file
        try (ObjectOutputStream oos = new ObjectOutputStream(new FileOutputStream(FILE_NAME))) {
            oos.writeObject(donors);
        } catch (IOException e) {
            e.printStackTrace();
        }

        // Read donors from file and display eligible ones
        try (ObjectInputStream ois = new ObjectInputStream(new FileInputStream(FILE_NAME))) {
            List<Donor> readDonors = (List<Donor>) ois.readObject();
            LocalDate sixMonthsAgo = LocalDate.now().minusMonths(6);

            System.out.println("\nEligible Donors (A+ve and not donated in the last 6 months):")
;
            for (Donor donor : readDonors) {
                if ("A+ve".equals(donor.getBloodGroup()) && donor.getLastDonationDate()
.isBefore(sixMonthsAgo)) {
                    System.out.println(donor);
                }
            }
        } catch (IOException | ClassNotFoundException e) {
            e.printStackTrace();
        }

        scanner.close();
```

```
    }
}

// output =
// Enter number of donors: 2
// Enter details for donor 1:
// Name: Rohit
// Age: 25
// Address: pune
// Contact Number: 2154873265
// Blood Group: A+ve
// Last Donation Date (yyyy-mm-dd): 2023-01-01
// Enter details for donor 2:
// Name: rohan
// Age: 30
// Address: mumbai
// Contact Number: 9865322154
// Blood Group: B+ve
// Last Donation Date (yyyy-mm-dd): 2024-09-20

// Eligible Donors (A+ve and not donated in the last 6 months):
// Name: Rohit, Age: 25, Address: pune, Contact: 2154873265, Blood Group: A+ve, Last Donation
Date: 2023-01-01
```

slip27_1.java

```java
// Define an Employee class with suitable attributes having getSalary() method, which returns
// salary withdrawn by a particular employee. Write a class Manager which extends a class
// Employee, override the getSalary() method, which will return salary of manager by adding
// traveling allowance, house rent allowance etc.


class Employee {
    String name;
    double basicSalary;

    public Employee(String name, double basicSalary) {
        this.name = name;
        this.basicSalary = basicSalary;
    }

    public double getSalary() {
        return basicSalary;
    }
}

class Manager extends Employee {
    double travelAllowance;
    double houseRentAllowance;

    public Manager(String name, double basicSalary, double travelAllowance, double
houseRentAllowance) {
        super(name, basicSalary);
        this.travelAllowance = travelAllowance;
        this.houseRentAllowance = houseRentAllowance;
    }

    @Override
    public double getSalary() {
        return basicSalary + travelAllowance + houseRentAllowance;
    }
}

public class slip27_1 {
    public static void main(String[] args) {
        Employee emp = new Employee("John Doe", 50000);
        Manager mgr = new Manager("Jane Smith", 60000, 10000, 15000);

        System.out.println("Employee Salary: " + emp.getSalary());
        System.out.println("Manager Salary: " + mgr.getSalary());
    }
}

// output =
// Employee Salary: 50000.0
// Manager Salary: 85000.0
```

slip27_2.java

```
// Write a program to accept a string as command line argument and check whether it is a file or
// directory. Also perform operations as follows:
// DIf it is a directory,delete all text files in that directory. Confirm delete operation from
// user before deleting text files. Also, display a count showing the number of files deleted,
// if any, from the directory.
// iD)If it is a file display various details of that file.


import java.io.File;
import java.util.Scanner;

public class slip27_2 {
    public static void main(String[] args) {
        if (args.length != 1) {
            System.out.println("Usage: java FileDirectoryCheck <file_or_directory_path>");
            return;
        }

        String path = args[0];
        File file = new File(path);

        if (file.exists()) {
            if (file.isDirectory()) {
                deleteTextFiles(file);
            } else if (file.isFile()) {
                displayFileDetails(file);
            } else {
                System.out.println("The path is neither a file nor a directory.");
            }
        } else {
            System.out.println("The specified path does not exist.");
        }
    }

    private static void deleteTextFiles(File directory) {
        File[] files = directory.listFiles();
        if (files == null || files.length == 0) {
            System.out.println("The directory is empty or not accessible.");
            return;
        }

        int deletedCount = 0;
        Scanner scanner = new Scanner(System.in);

        System.out.println("The following text files will be deleted:");
        for (File file : files) {
            if (file.isFile() && file.getName().endsWith(".txt")) {
                System.out.println(file.getName());
            }
        }
```

```
        System.out.print("Do you want to delete these files? (yes/no): ");
        String confirmation = scanner.nextLine();

        if (confirmation.equalsIgnoreCase("yes")) {
            for (File file : files) {
                if (file.isFile() && file.getName().endsWith(".txt")) {
                    if (file.delete()) {
                        deletedCount++;
                    } else {
                        System.out.println("Failed to delete: " + file.getName());
                    }
                }
            }
            System.out.println(deletedCount + " text files deleted from the directory.");
        } else {
            System.out.println("Deletion canceled.");
        }

        scanner.close();
    }

    private static void displayFileDetails(File file) {
        System.out.println("File Name: " + file.getName());
        System.out.println("Absolute Path: " + file.getAbsolutePath());
        System.out.println("File Size: " + file.length() + " bytes");
        System.out.println("Writable: " + file.canWrite());
        System.out.println("Readable: " + file.canRead());
        System.out.println("Executable: " + file.canExecute());
    }
}
```

slip29_1.java

```
import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;

class Customer {
    int custNo;
    String custName;
    String contactNumber;
    String custAddr;

    public Customer(int custNo, String custName, String contactNumber, String custAddr) {
        this.custNo = custNo;
        this.custName = custName;
        this.contactNumber = contactNumber;
        this.custAddr = custAddr;
    }
}

public class slip29_1 {
    private List<Customer> customers;

    public slip29_1() {
        customers = new ArrayList<>();
        customers.add(new Customer(1, "John Doe", "9876543210", "123 Main St"));
        customers.add(new Customer(2, "Jane Smith", "9876543211", "456 Elm St"));
        customers.add(new Customer(3, "Alice Johnson", "9876543212", "789 Pine St"));
    }

    public void searchCustomer(String contactNumber) {
        for (Customer c : customers) {
            if (c.contactNumber.equals(contactNumber)) {
                System.out.println("Customer Number: " + c.custNo);
                System.out.println("Customer Name: " + c.custName);
                System.out.println("Contact Number: " + c.contactNumber);
                System.out.println("Customer Address: " + c.custAddr);
                return;
            }
        }
        System.out.println("Customer not found.");
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        slip29_1 cm = new slip29_1();

        System.out.print("Enter contact number to search: ");
        String contactNumber = scanner.nextLine();
        cm.searchCustomer(contactNumber);

        scanner.close();
    }
```

```
}

// output =
// Enter contact number to search: 9876543210
// Customer Number: 1
// Customer Name: John Doe
// Contact Number: 9876543210
// Customer Address: 123 Main St
```

slip29_2.java

```
// Write a program to create a super class Vehicle having members Company and price.
// Derive two different classes LightMotorVehicle(mileage) and HeavyMotorVehicle
// (capacity_in_tons). Accept the information for "n" vehicles and display the information in
// appropriate form. While taking data, ask user about the type of vehicle first.


import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;

class Vehicle {
    String c; // Company
    double p; // Price

    public Vehicle(String c, double p) {
        this.c = c;
        this.p = p;
    }

    public void displayInfo() {
        System.out.println("Company: " + c + ", Price: $" + p);
    }
}

class LightMotorVehicle extends Vehicle {
    double m; // Mileage

    public LightMotorVehicle(String c, double p, double m) {
        super(c, p);
        this.m = m;
    }

    @Override
    public void displayInfo() {
        super.displayInfo();
        System.out.println("Mileage: " + m + " km/l");
    }
}

class HeavyMotorVehicle extends Vehicle {
    double cap; // Capacity in tons

    public HeavyMotorVehicle(String c, double p, double cap) {
        super(c, p);
        this.cap = cap;
    }

    @Override
    public void displayInfo() {
        super.displayInfo();
        System.out.println("Capacity: " + cap + " tons");
```

```
    }
}

public class slip29_2 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        List<Vehicle> vList = new ArrayList<>(); // Vehicle list

        System.out.print("Enter the number of vehicles: ");
        int n = sc.nextInt();
        sc.nextLine(); // Consume newline

        for (int i = 0; i < n; i++) {
            System.out.print("Enter vehicle type (Light/Heavy): ");
            String type = sc.nextLine().trim().toLowerCase();
            System.out.print("Enter Company Name: ");
            String c = sc.nextLine(); // Company name
            System.out.print("Enter Price: ");
            double p = sc.nextDouble(); // Price

            Vehicle v = null; // Vehicle reference
            if (type.equals("light")) {
                System.out.print("Enter Mileage (km/l): ");
                double m = sc.nextDouble(); // Mileage
                v = new LightMotorVehicle(c, p, m);
            } else if (type.equals("heavy")) {
                System.out.print("Enter Capacity (tons): ");
                double cap = sc.nextDouble(); // Capacity
                v = new HeavyMotorVehicle(c, p, cap);
            } else {
                System.out.println("Invalid vehicle type! Please enter 'Light' or 'Heavy'.");
                i--; // Retry this vehicle entry
            }

            if (v != null) {
                vList.add(v);
            }
            sc.nextLine(); // Consume newline
        }

        System.out.println("\nVehicle Information:");
        for (Vehicle v : vList) {
            v.displayInfo();
            System.out.println("--------------------------");
        }

        sc.close();
    }
}

// output =
// Enter the number of vehicles: 2
// Enter vehicle type (Light/Heavy): light
// Enter Company Name: toyato
```

```
// Enter Price: 500000
// Enter Mileage (km/l): 15
// Enter vehicle type (Light/Heavy): heavy
// Enter Company Name: volva
// Enter Price: 200000
// Enter Capacity (tons): 20

// Vehicle Information:
// Company: toyato, Price: $500000.0
// Mileage: 15.0 km/l
// --------------------------
// Company: volva, Price: $200000.0
// Capacity: 20.0 tons
// --------------------------
```

slip30_1.java

```java
// Write program to define class Person with data member as Personname,Aadharno, Panno.
// Accept information for 5 objects and display appropriate information (use this keyword).

import java.util.Scanner;

class Person {
    String personName;
    String aadharNo;
    String panNo;

    // Constructor to initialize Person attributes
    public Person(String personName, String aadharNo, String panNo) {
        this.personName = personName; // Using 'this' keyword
        this.aadharNo = aadharNo;     // Using 'this' keyword
        this.panNo = panNo;           // Using 'this' keyword
    }

    // Method to display person information
    public void displayInfo() {
        System.out.println("Name: " + personName);
        System.out.println("Aadhar No: " + aadharNo);
        System.out.println("PAN No: " + panNo);
        System.out.println("----------------------------");
    }
}

public class slip30_1 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        Person[] persons = new Person[5]; // Array to store 5 Person objects

        // Accept information for 5 Person objects
        for (int i = 0; i < 5; i++) {
            System.out.print("Enter Person Name: ");
            String name = scanner.nextLine();
            System.out.print("Enter Aadhar No: ");
            String aadhar = scanner.nextLine();
            System.out.print("Enter PAN No: ");
            String pan = scanner.nextLine();

            persons[i] = new Person(name, aadhar, pan); // Creating a new Person object
        }

        System.out.println("\nPerson Information:");
        // Display information for each Person object
        for (Person person : persons) {
            person.displayInfo();
        }

        scanner.close();
    }
```

```
}

// output =
// Enter Person Name: John Doe
// Enter Aadhar No: 1234-5678-9012
// Enter PAN No: ABCDE1234F
// Enter Person Name: Jane Smith
// Enter Aadhar No: 9876-5432-1098
// Enter PAN No: WXYZT5678G
// Enter Person Name: Alice Johnson
// Enter Aadhar No: 5678-1234-4567
// Enter PAN No: PQRSF2345H
// Enter Person Name: Bob Brown
// Enter Aadhar No: 4321-8765-6789
// Enter PAN No: LMNOP6789J
// Enter Person Name: Charlie Black
// Enter Aadhar No: 3456-7890-1234
// Enter PAN No: UVWXY2345K

// Person Information:
// Name: John Doe
// Aadhar No: 1234-5678-9012
// PAN No: ABCDE1234F
// ----------------------------
// Name: Jane Smith
// Aadhar No: 9876-5432-1098
// PAN No: WXYZT5678G
// ----------------------------
// Name: Alice Johnson
// Aadhar No: 5678-1234-4567
// PAN No: PQRSF2345H
// ----------------------------
// Name: Bob Brown
// Aadhar No: 4321-8765-6789
// PAN No: LMNOP6789J
// ----------------------------
// Name: Charlie Black
// Aadhar No: 3456-7890-1234
// PAN No: UVWXY2345K
// ----------------------------
```

slip30_2.java

```
// Write a program that creates a user interface to perform integer divisions. The user enters two
// numbers in the text fields, Numberl and Number2. The division of Numberl and Number2 is
// displayed in the Result field when the Divide button is clicked. If Numberl or Number2 were
// not an integer, the program would throw a NumberFormatException. If Number2 were Zero,
// the program would throw an Arithmetic Exception Display the exception in a message
// dialog box.


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


public class slip30_2 {
    public static void main(String[] args) {
        JFrame f = new JFrame("Int Division");
        f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        f.setSize(300, 150);
        f.setLayout(new FlowLayout());

        JTextField num1Field = new JTextField(10);
        JTextField num2Field = new JTextField(10);
        JLabel resultLabel = new JLabel("Result:");

        JButton divBtn = new JButton("Divide");
        divBtn.addActionListener(e -> {
            try {
                int n1 = Integer.parseInt(num1Field.getText());
                int n2 = Integer.parseInt(num2Field.getText());
                resultLabel.setText("Result: " + (n1 / n2));
            } catch (NumberFormatException ex) {
```

```
                JOptionPane.showMessageDialog(f, "Enter valid
integers.", "Input Error", JOptionPane.ERROR_MESSAGE);
            } catch (ArithmeticException ex) {
                JOptionPane.showMessageDialog(f, "Division by
zero not allowed.", "Math Error", JOptionPane.ERROR_MESSAGE);
            }
        });

        f.add(new JLabel("Number 1:"));
        f.add(num1Field);
        f.add(new JLabel("Number 2:"));
        f.add(num2Field);
        f.add(divBtn);
        f.add(resultLabel);

        f.setVisible(true);
    }
}
```