

Assignment-IV

Name:- Bibek Chand Sah

Roll:- 22054029

Section:- CSE-05

1. Given are two one-dimensional arrays A & B, which are sorted in ascending order. Write a Java program to merge them into single sorted array C that contains every item from arrays A & B, in ascending order.



```
import java.util.Arrays;

public class MergeSortedArrays {
    public static void main(String[] args) {
        int[] A = { 1, 3, 5, 7, 9 };
        int[] B = { 2, 4, 6, 8, 10 };

        int[] C = new int[A.length + B.length];
        int i = 0, j = 0, k = 0;

        while (i < A.length && j < B.length) {
            if (A[i] < B[j]) {
                C[k++] = A[i++];
            } else {
                C[k++] = B[j++];
            }
        }

        while (i < A.length) {
            C[k++] = A[i++];
        }

        while (j < B.length) {
            C[k++] = B[j++];
        }

        System.out.println("Merged Array C: " + Arrays.toString(C));
    }
}
```

Merged Array C: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

2. Write a Java program to show 0-arguments constructor.



```
public class ZeroArgumentsConstructor {
    public ZeroArgumentsConstructor() {
        System.out.println("This is a 0-arguments constructor.");
    }

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

```

        ZeroArgumentsConstructor obj = new ZeroArgumentsConstructor();
    }
}

```

This is a 0-arguments constructor.

3. Write a Java program to show parameterized constructor.



```

public class ParameterizedConstructor {
    private int value;

    public ParameterizedConstructor(int val) {
        value = val;
    }

    public void displayValue() {
        System.out.println("Value: " + value);
    }

    public static void main(String[] args) {
        ParameterizedConstructor obj = new ParameterizedConstructor(10);
        obj.displayValue();
    }
}

```

Value: 10

4. Write a Java program to show constructor overloading.



```

public class ConstructorOverloading {
    private int value;

    public ConstructorOverloading() {
        value = 0;
    }

    public ConstructorOverloading(int val) {
        value = val;
    }

    public void displayValue() {
        System.out.println("Value: " + value);
    }

    public static void main(String[] args) {

```

```

        ConstructorOverloading obj1 = new ConstructorOverloading();
        ConstructorOverloading obj2 = new ConstructorOverloading(10);

        obj1.displayValue();
        obj2.displayValue();
    }
}

```

Value: 0

Value: 10

5. Write a class, Grader, which has an instance variable, score, an appropriate constructor and appropriate methods. A method, letterGrade() that returns the letter grade as O/E/A/B/C/F. Now write a demo class to test the Grader class by reading a score from the user, using it to create a Grader object after validating that the value is not negative and is not greater than 100. Finally, call the letterGrade() method to get and print the grade.

→

```

import java.util.Scanner;

class Grader {
    private int score;

    public Grader(int score) {
        this.score = score;
    }

    public String letterGrade() {
        if (score >= 90 && score <= 100) {
            return "O";
        } else if (score >= 80 && score < 90) {
            return "E";
        } else if (score >= 70 && score < 80) {
            return "A";
        } else if (score >= 60 && score < 70) {
            return "B";
        } else if (score >= 50 && score < 60) {
            return "C";
        } else {
            return "F";
        }
    }
}

public class GraderDemo {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the score: ");
    }
}

```

```

        int score = scanner.nextInt();

        if (score >= 0 && score <= 100) {
            Grader grader = new Grader(score);
            System.out.println("Grade: " + grader.letterGrade());
        } else {
            System.out.println("Invalid score! Score must be between 0 and
100.");
        }

        scanner.close();
    }
}

```

Enter the score: 91

Grade: O

6. Write a class, Commission, which has an instance variable, sales; an appropriate constructor; and a method, commission() that returns the commission.

Now write a demo class to test the Commission class by reading a sale from the user, using it to create a Commission object after validating that the value is not negative.

Finally, call the commission() method to get and print the commission. If the sales are negative, your demo should print the message “Invalid Input”.

➔

```

import java.util.Scanner;

class Commission {
    private double sales;

    public Commission(double sales) {
        this.sales = sales;
    }

    public double calculateCommission() {
        if (sales < 0) {
            System.out.println("Invalid Input! Sales cannot be negative.");
            return -1;
        } else {
            return sales * 0.10; // Assuming commission rate is 10%
        }
    }
}

```

```

public class CommissionDemo {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the sales amount: ");
        double sales = scanner.nextDouble();

        Commission commission = new Commission(sales);
        double commissionAmount = commission.calculateCommission();

        if (commissionAmount != -1) {
            System.out.println("Commission: " + commissionAmount);
        }

        scanner.close();
    }
}

```

Enter the sales amount: 20000

Commission: 2000.0

6a. Create a general class ThreeDObject and derive the classes Box, Cube, Cylinder and Cone from it. The class ThreeDObject has methods wholeSurfaceArea () and volume (). Override these two methods in each of the derived classes to calculate the volume and whole surface area of each type of three-dimensional objects. The dimensions of the objects are to be taken from the users and passed through the respective constructors of each derived class. Write a main method to test these classes.

➔

```

import java.util.Scanner;

class ThreeDObject {
    protected double length;
    protected double width;
    protected double height;

    public ThreeDObject(double length, double width, double height) {
        this.length = length;
        this.width = width;
        this.height = height;
    }

    public double wholeSurfaceArea() {
        return 2 * (length * width + length * height + width * height);
    }
}

```

```

        public double volume() {
            return length * width * height;
        }
    }

class Box extends ThreeDObject {
    public Box(double length, double width, double height) {
        super(length, width, height);
    }

    @Override
    public double wholeSurfaceArea() {
        return 2 * (length * width + length * height + width * height);
    }

    @Override
    public double volume() {
        return length * width * height;
    }
}

class Cube extends ThreeDObject {
    public Cube(double side) {
        super(side, side, side);
    }
}

class Cylinder extends ThreeDObject {
    public Cylinder(double radius, double height) {
        super(radius, radius, height);
    }

    @Override
    public double wholeSurfaceArea() {
        return 2 * Math.PI * length * (length + height);
    }

    @Override
    public double volume() {
        return Math.PI * length * length * height;
    }
}

class Cone extends ThreeDObject {
    public Cone(double radius, double height) {
        super(radius, height, 0);
    }
}

```

```

@Override
public double wholeSurfaceArea() {
    double slantHeight = Math.sqrt(length * length + height * height);
    return Math.PI * length * (length + slantHeight);
}

@Override
public double volume() {
    return (1.0 / 3.0) * Math.PI * length * length * height;
}
}

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

        System.out.println("Enter dimensions for a Box (length, width, height):");
        double boxLength = scanner.nextDouble();
        double boxWidth = scanner.nextDouble();
        double boxHeight = scanner.nextDouble();
        Box box = new Box(boxLength, boxWidth, boxHeight);
        System.out.println("Box Surface Area: " + box.wholeSurfaceArea());
        System.out.println("Box Volume: " + box.volume());

        System.out.println("\nEnter side length for a Cube:");
        double cubeSide = scanner.nextDouble();
        Cube cube = new Cube(cubeSide);
        System.out.println("Cube Surface Area: " + cube.wholeSurfaceArea());
        System.out.println("Cube Volume: " + cube.volume());

        System.out.println("\nEnter radius and height for a Cylinder:");
        double cylinderRadius = scanner.nextDouble();
        double cylinderHeight = scanner.nextDouble();
        Cylinder cylinder = new Cylinder(cylinderRadius, cylinderHeight);
        System.out.println("Cylinder Surface Area: " +
cylinder.wholeSurfaceArea());
        System.out.println("Cylinder Volume: " + cylinder.volume());

        System.out.println("\nEnter radius and height for a Cone:");
        double coneRadius = scanner.nextDouble();
        double coneHeight = scanner.nextDouble();
        Cone cone = new Cone(coneRadius, coneHeight);
        System.out.println("Cone Surface Area: " + cone.wholeSurfaceArea());
        System.out.println("Cone Volume: " + cone.volume());

        scanner.close();
    }
}

```



```
}  
}
```

Enter dimensions for a Box (length, width, height):

5 6 7

Box Surface Area: 214.0

Box Volume: 210.0

Enter side length for a Cube:

5

Cube Surface Area: 150.0

Cube Volume: 125.0

Enter radius and height for a Cylinder:

5 6

Cylinder Surface Area: 345.57519189487726

Cylinder Volume: 471.23889803846896

Enter radius and height for a Cone:

5 6

Cone Surface Area: 157.07963267948966

Cone Volume: 0.0

6b. Write a program to create a class named Vehicle having protected instance variables regnNumber, speed, color, ownerName and a method showData () to show “This is a vehicle class”. Inherit the Vehicle class into subclasses named Bus and Car having individual private instance variables routeNumber in Bus and manufacturerName in Car and both of them having showData () method showing all details of Bus and Car respectively with content of the super class’s showData () method.

→

```
import java.util.Scanner;  
  
class Vehicle {  
    protected String regnNumber;  
    protected int speed;  
    protected String color;  
    protected String ownerName;
```

```
    public Vehicle(String regnNumber, int speed, String color, String
ownerName) {
        this.regnNumber = regnNumber;
        this.speed = speed;
        this.color = color;
        this.ownerName = ownerName;
    }

    public void showData() {
        System.out.println("Registration Number: " + regnNumber);
        System.out.println("Speed: " + speed);
        System.out.println("Color: " + color);
        System.out.println("Owner Name: " + ownerName);
    }
}

class Car extends Vehicle {
    private int numberOfPassengers;

    public Car(String regnNumber, int speed, String color, String ownerName,
int numberOfPassengers) {
        super(regnNumber, speed, color, ownerName);
        this.numberOfPassengers = numberOfPassengers;
    }

    @Override
    public void showData() {
        super.showData();
        System.out.println("Number of Passengers: " + numberOfPassengers);
    }
}

class Truck extends Vehicle {
    private int loadLimit;

    public Truck(String regnNumber, int speed, String color, String ownerName,
int loadLimit) {
        super(regnNumber, speed, color, ownerName);
        this.loadLimit = loadLimit;
    }

    @Override
    public void showData() {
        super.showData();
        System.out.println("Load Limit: " + loadLimit);
    }
}
```

```

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

        System.out.println("Enter details for a car:");
        System.out.print("Registration Number: ");
        String carRegnNumber = scanner.nextLine();
        System.out.print("Speed: ");
        int carSpeed = scanner.nextInt();
        scanner.nextLine(); // Consume newline character
        System.out.print("Color: ");
        String carColor = scanner.nextLine();
        System.out.print("Owner Name: ");
        String carOwnerName = scanner.nextLine();
        System.out.print("Number of Passengers: ");
        int carPassengers = scanner.nextInt();

        Car car = new Car(carRegnNumber, carSpeed, carColor, carOwnerName,
carPassengers);

        System.out.println("\nEnter details for a truck:");
        scanner.nextLine(); // Consume newline character
        System.out.print("Registration Number: ");
        String truckRegnNumber = scanner.nextLine();
        System.out.print("Speed: ");
        int truckSpeed = scanner.nextInt();
        scanner.nextLine(); // Consume newline character
        System.out.print("Color: ");
        String truckColor = scanner.nextLine();
        System.out.print("Owner Name: ");
        String truckOwnerName = scanner.nextLine();
        System.out.print("Load Limit: ");
        int truckLoadLimit = scanner.nextInt();

        Truck truck = new Truck(truckRegnNumber, truckSpeed, truckColor,
truckOwnerName, truckLoadLimit);

        System.out.println("\nCar Details:");
        car.showData();

        System.out.println("\nTruck Details:");
        truck.showData();

        scanner.close();
    }
}

```

Enter details for a car:

Registration Number: bib12

Speed: 120
Color: black
Owner Name: bibek
Number of Passengers: 5

Enter details for a truck:
Registration Number: bib13
Speed: 110
Color: red & black
Owner Name: bibek
Load Limit: 300

Car Details:
Registration Number: bib12
Speed: 120
Color: black
Owner Name: bibek
Number of Passengers: 5

Truck Details:
Registration Number: bib13
Speed: 110
Color: red & black
Owner Name: bibek
Load Limit: 300

7. An educational institution maintains a database of its employees. The database is divided into a number of classes whose hierarchical relationships are shown below.

Write all the classes and define the methods to create the database and retrieve individual information as and when needed.

Write a driver program to test the classes.

Staff (code, name) Teacher (subject, publication) is a Staff

Officer (grade) is a Staff Typist (speed) is a Staff

RegularTypist (remuneration) is a Typist CasualTypist (daily wages) is a Typist.



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

class Staff {
    protected int code;
    protected String name;
```

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

public void displayInfo() {
    System.out.println("Code: " + code);
    System.out.println("Name: " + name);
}
}

class Teacher extends Staff {
    private String subject;
    private String publication;

    public Teacher(int code, String name, String subject, String publication)
    {
        super(code, name);
        this.subject = subject;
        this.publication = publication;
    }

    @Override
    public void displayInfo() {
        super.displayInfo();
        System.out.println("Subject: " + subject);
        System.out.println("Publication: " + publication);
    }
}

class Officer extends Staff {
    private String grade;

    public Officer(int code, String name, String grade) {
        super(code, name);
        this.grade = grade;
    }

    @Override
    public void displayInfo() {
        super.displayInfo();
        System.out.println("Grade: " + grade);
    }
}

class Typist extends Staff {
    private int speed;
}
```

```

    public Typist(int code, String name, int speed) {
        super(code, name);
        this.speed = speed;
    }

    @Override
    public void displayInfo() {
        super.displayInfo();
        System.out.println("Speed: " + speed);
    }
}

class RegularTypist extends Typist {
    private double remuneration;

    public RegularTypist(int code, String name, int speed, double remuneration) {
        super(code, name, speed);
        this.remuneration = remuneration;
    }

    @Override
    public void displayInfo() {
        super.displayInfo();
        System.out.println("Remuneration: " + remuneration);
    }
}

class CasualTypist extends Typist {
    private double dailyWages;

    public CasualTypist(int code, String name, int speed, double dailyWages) {
        super(code, name, speed);
        this.dailyWages = dailyWages;
    }

    @Override
    public void displayInfo() {
        super.displayInfo();
        System.out.println("Daily Wages: " + dailyWages);
    }
}

public class EducationalInstitutionDatabase {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        ArrayList<Staff> staffList = new ArrayList<>();
    }
}

```

```

        System.out.println("Enter staff details:");
        System.out.print("Number of staff: ");
        int numStaff = scanner.nextInt();
        scanner.nextLine(); // Consume newline character

        for (int i = 0; i < numStaff; i++) {
            System.out.println("\nEnter details for staff #" + (i + 1));
            System.out.print("Code: ");
            int code = scanner.nextInt();
            scanner.nextLine(); // Consume newline character
            System.out.print("Name: ");
            String name = scanner.nextLine();
            System.out.print("Type (1. Teacher, 2. Officer, 3. Regular Typist,
4. Casual Typist): ");
            int type = scanner.nextInt();
            scanner.nextLine(); // Consume newline character

            switch (type) {
                case 1:
                    System.out.print("Subject: ");
                    String subject = scanner.nextLine();
                    System.out.print("Publication: ");
                    String publication = scanner.nextLine();
                    staffList.add(new Teacher(code, name, subject,
publication));
                    break;
                case 2:
                    System.out.print("Grade: ");
                    String grade = scanner.nextLine();
                    staffList.add(new Officer(code, name, grade));
                    break;
                case 3:
                    System.out.print("Speed: ");
                    int speed1 = scanner.nextInt();
                    System.out.print("Remuneration: ");
                    double remuneration = scanner.nextDouble();
                    staffList.add(new RegularTypist(code, name, speed1,
remuneration));
                    break;
                case 4:
                    System.out.print("Speed: ");
                    int speed2 = scanner.nextInt();
                    System.out.print("Daily Wages: ");
                    double dailyWages = scanner.nextDouble();
                    staffList.add(new CasualTypist(code, name, speed2,
dailyWages));
                    break;
                default:

```

```

        System.out.println("Invalid staff type!");
        break;
    }
}

System.out.println("\nStaff Details:");
for (Staff staff : staffList) {
    staff.displayInfo();
    System.out.println();
}

scanner.close();
}
}

```

Enter staff details:

Number of staff: 2

Enter details for staff #1

Code: 4029

Name: bibek

Type (1. Teacher, 2. Officer, 3. Regular Typist, 4. Casual Typist): 2

Grade: A

Enter details for staff #2

Code: 4030

Name: ram

Type (1. Teacher, 2. Officer, 3. Regular Typist, 4. Casual Typist): 1

Subject: math

Publication: ekta

Staff Details:

Code: 4029

Name: bibek

Grade: A

Code: 4030

Name: ram

Subject: math

Publication: ekta

8. Create a base class Building that stores the number of floors of a building, number of rooms and it's total footage. Create a derived class House that inherits Building and also stores the number of bedrooms and bathrooms. Demonstrate the working of the classes.




```
import java.util.Scanner;

class Building {
    protected int numberOfFloors;
    protected int numberOfRooms;
    protected double totalFootage;

    public Building(int numberOfFloors, int numberOfRooms, double
totalFootage) {
        this.numberOfFloors = numberOfFloors;
        this.numberOfRooms = numberOfRooms;
        this.totalFootage = totalFootage;
    }

    public void displayDetails() {
        System.out.println("Number of Floors: " + numberOfFloors);
        System.out.println("Number of Rooms: " + numberOfRooms);
        System.out.println("Total Footage: " + totalFootage);
    }
}

class House extends Building {
    private int numberOfBedrooms;
    private int numberOfBathrooms;

    public House(int numberOfFloors, int numberOfRooms, double totalFootage,
int numberOfBedrooms, int numberOfBathrooms) {
        super(numberOfFloors, numberOfRooms, totalFootage);
        this.numberOfBedrooms = numberOfBedrooms;
        this.numberOfBathrooms = numberOfBathrooms;
    }

    @Override
    public void displayDetails() {
        super.displayDetails();
        System.out.println("Number of Bedrooms: " + numberOfBedrooms);
        System.out.println("Number of Bathrooms: " + numberOfBathrooms);
    }
}

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

        System.out.println("Enter details for a building:");
        System.out.print("Number of Floors: ");
        int numOfFloors = scanner.nextInt();
        System.out.print("Number of Rooms: ");
```

```

        int numOfRooms = scanner.nextInt();
        System.out.print("Total Footage: ");
        double totalFootage = scanner.nextDouble();

        Building building = new Building(numOfFloors, numOfRooms,
totalFootage);

        System.out.println("\nEnter details for a house:");
        System.out.print("Number of Floors: ");
        int houseFloors = scanner.nextInt();
        System.out.print("Number of Rooms: ");
        int houseRooms = scanner.nextInt();
        System.out.print("Total Footage: ");
        double houseFootage = scanner.nextDouble();
        System.out.print("Number of Bedrooms: ");
        int numOfBedrooms = scanner.nextInt();
        System.out.print("Number of Bathrooms: ");
        int numOfBathrooms = scanner.nextInt();

        House house = new House(houseFloors, houseRooms, houseFootage,
numOfBedrooms, numOfBathrooms);

        System.out.println("\nBuilding Details:");
        building.displayDetails();

        System.out.println("\nHouse Details:");
        house.displayDetails();

        scanner.close();
    }
}

```

Enter details for a building:

Number of Floors: 3

Number of Rooms: 5

Total Footage: 8

Enter details for a house:

Number of Floors: 2

Number of Rooms: 4

Total Footage: 6

Number of Bedrooms: 3

Number of Bathrooms: 3

Building Details:

Number of Floors: 3

Number of Rooms: 5

Total Footage: 8.0

House Details:

Number of Floors: 2

Number of Rooms: 4

Total Footage: 6.0

Number of Bedrooms: 3

Number of Bathrooms: 3

9. In the earlier program, create a second derived class Office that inherits Building and stores the number of telephones and tables. Now demonstrate the working of all three classes.



```
import java.util.Scanner;

class Office extends Building {
    private int numberOfTelephones;
    private int numberOfTables;

    public Office(int numberOfFloors, int numberOfRooms, double totalFootage,
int numberOfTelephones,
        int numberOfTables) {
        super(numberOfFloors, numberOfRooms, totalFootage);
        this.numberOfTelephones = numberOfTelephones;
        this.numberOfTables = numberOfTables;
    }

    @Override
    public void displayDetails() {
        super.displayDetails();
        System.out.println("Number of Telephones: " + numberOfTelephones);
        System.out.println("Number of Tables: " + numberOfTables);
    }
}

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

        System.out.println("Enter details for an office:");
        System.out.print("Number of Floors: ");
        int officeFloors = scanner.nextInt();
        System.out.print("Number of Rooms: ");
        int officeRooms = scanner.nextInt();
        System.out.print("Total Footage: ");
        double officeFootage = scanner.nextDouble();
        System.out.print("Number of Telephones: ");
```

```

        int numOfTelephones = scanner.nextInt();
        System.out.print("Number of Tables: ");
        int numOfTables = scanner.nextInt();

        Office office = new Office(officeFloors, officeRooms, officeFootage,
numOfTelephones, numOfTables);

        System.out.println("\nOffice Details:");
        office.displayDetails();

        scanner.close();
    }
}

```

Enter details for an office:

Number of Floors: 3

Number of Rooms: 5

Total Footage: 6

Number of Telephones: 3

Number of Tables: 5

Office Details:

Number of Floors: 3

Number of Rooms: 5

Total Footage: 6.0

Number of Telephones: 3

Number of Tables: 5

10. Write a Java program which creates a base class Num and contains an integer number along with a method shownum() which displays the number. Now create a derived class HexNum which inherits Num and overrides shownum() which displays the hexadecimal value of the number. Demonstrate the working of the classes.



```

import java.util.Scanner;

class Num {
    protected int number;

    public Num(int number) {
        this.number = number;
    }

    public void showNum() {
        System.out.println("Number: " + number);
    }
}

```

```

}

class HexNum extends Num {
    public HexNum(int number) {
        super(number);
    }

    @Override
    public void showNum() {
        System.out.println("Hexadecimal Value: " +
Integer.toHexString(number));
    }
}

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

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

        Num numObj = new Num(num);
        HexNum hexNumObj = new HexNum(num);

        System.out.println("\nNumber:");
        numObj.showNum();

        System.out.println("\nHexadecimal Value:");
        hexNumObj.showNum();

        scanner.close();
    }
}

```

Enter a number: 15

Number:

Number: 15

Hexadecimal Value:

Hexadecimal Value: f

11. Write a Java program which creates a base class Num and contains an integer number along with a method shownum() which displays the number. Now create a derived class OctNum which inherits Num and overrides shownum() which displays the octal value of the number. Demonstrate the working of the classes.



```
import java.util.Scanner;

class OctNum extends Num {
    public OctNum(int number) {
        super(number);
    }

    @Override
    public void showNum() {
        System.out.println("Octal Value: " + Integer.toOctalString(number));
    }
}

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

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

        Num numObj = new Num(num);
        OctNum octNumObj = new OctNum(num);

        System.out.println("\nNumber:");
        numObj.showNum();

        System.out.println("\nOctal Value:");
        octNumObj.showNum();

        scanner.close();
    }
}
```

Enter a number: 12

Number:

Number: 12

Octal Value:

Octal Value: 14

12. Combine Question number 10 and 11 and have all the three classes together. Now describe the working of all classes.



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

```

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

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

        Num numObj = new Num(num);
        HexNum hexNumObj = new HexNum(num);
        OctNum octNumObj = new OctNum(num);

        System.out.println("\nNumber:");
        numObj.showNum();

        System.out.println("\nHexadecimal Value:");
        hexNumObj.showNum();

        System.out.println("\nOctal Value:");
        octNumObj.showNum();

        scanner.close();
    }
}

```

Enter a number: 12

Number:

Number: 12

Hexadecimal Value:

Hexadecimal Value: c

Octal Value:

Octal Value: 14

13. Create a base class Distance which stores the distance between two locations in miles and a method travelTime(). The method prints the time taken to cover the distance when the speed is 60 miles per hour. Now in a derived class DistanceMKS, override travelTime() so that it prints the time assuming the distance is in kilometers and the speed is 100 km per second. Demonstrate the working of the classes.

→

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

```

class Distance {
    protected double distance;

    public Distance(double distance) {
        this.distance = distance;
    }

    public void travelTime() {
        double time = distance / 60; // Assuming speed is 60 miles per hour
        System.out.println("Time taken to cover the distance: " + time + "
hours");
    }
}

class DistanceMKS extends Distance {
    public DistanceMKS(double distance) {
        super(distance);
    }

    @Override
    public void travelTime() {
        double time = distance / 100; // Assuming speed is 100 km per hour
        System.out.println("Time taken to cover the distance: " + time + "
hours");
    }
}

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

        System.out.print("Enter distance in miles: ");
        double distanceMiles = scanner.nextDouble();
        Distance distanceObj = new Distance(distanceMiles);
        distanceObj.travelTime();

        System.out.print("\nEnter distance in kilometers: ");
        double distanceKm = scanner.nextDouble();
        DistanceMKS distanceMKSOBJ = new DistanceMKS(distanceKm);
        distanceMKSOBJ.travelTime();

        scanner.close();
    }
}

```

Enter distance in miles: 20

Time taken to cover the distance: 0.3333333333333333 hours

Enter distance in kilometers: 20
Time taken to cover the distance: 0.2 hours

14. Create a base class called “vehicle” that stores number of wheels and speed. Create the following derived classes – “car” that inherits “vehicle” and also stores number of passengers. “truck” that inherits “vehicle” and also stores the load limit. Write a main function to create objects of these two derived classes and display all the information about “car” and “truck”. Also compare the speed of these two vehicles - car and truck and display which one is faster

→

15. Aim of the program : A Plastic manufacturer sells plastic in different shapes like 2D sheet and 3D box. The cost of sheet is Rs 40/ per square ft. and the cost of box is Rs 60/ per cubic ft. Implement it in Java to calculate the cost of plastic as per the dimensions given by the user where 3D inherits from 2D.

→

```
import java.util.Scanner;

class Plastic {
    protected double costPerSquareFoot;

    public Plastic(double costPerSquareFoot) {
        this.costPerSquareFoot = costPerSquareFoot;
    }

    public double calculateCost(double area) {
        return area * costPerSquareFoot;
    }
}

class Sheet extends Plastic {
    public Sheet(double costPerSquareFoot) {
        super(costPerSquareFoot);
    }
}

class Box extends Plastic {
    public Box(double costPerCubicFoot) {
        super(costPerCubicFoot);
    }

    @Override
    public double calculateCost(double volume) {
        return volume * costPerSquareFoot;
    }
}
```

```

    }
}

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

        System.out.print("Enter dimensions for sheet (in square feet): ");
        double sheetArea = scanner.nextDouble();
        System.out.print("Enter cost per square foot: ");
        double sheetCost = scanner.nextDouble();
        Sheet sheet = new Sheet(sheetCost);
        double sheetTotalCost = sheet.calculateCost(sheetArea);
        System.out.println("Cost of plastic sheet: Rs. " + sheetTotalCost);

        System.out.print("\nEnter dimensions for box (in cubic feet): ");
        double boxVolume = scanner.nextDouble();
        System.out.print("Enter cost per cubic foot: ");
        double boxCost = scanner.nextDouble();
        Box box = new Box(boxCost);
        double boxTotalCost = box.calculateCost(boxVolume);
        System.out.println("Cost of plastic box: Rs. " + boxTotalCost);

        scanner.close();
    }
}

```

Enter dimensions for sheet (in square feet): 5

Enter cost per square foot: 20

Cost of plastic sheet: Rs. 100.0

Enter dimensions for box (in cubic feet): 6

Enter cost per cubic foot: 15

Cost of plastic box: Rs. 90.0

16. Aim of the program : Illustrate the execution of constructors in multi-level inheritance with three Java classes – plate(length, width), box(length, width, height), wood box (length, width, height, thick) where box inherits from plate and woodbox inherits from box class. Each class has constructor where dimensions are taken from user.



```

import java.util.Scanner;

class ThreeDObject {
    protected double length;
    protected double width;
    protected double height;
}

```

```

    public ThreeDObject(double length, double width, double height) {
        this.length = length;
        this.width = width;
        this.height = height;
    }

    public double wholeSurfaceArea() {
        return 2 * (length * width + length * height + width * height);
    }

    public double volume() {
        return length * width * height;
    }
}

class Plate extends ThreeDObject {
    public Plate(double length, double width) {
        super(length, width, 0); // Height is considered as 0 for plate
    }
}

class Box extends Plate {
    public Box(double length, double width, double height) {
        super(length, width);
        this.height = height;
    }

    @Override
    public double wholeSurfaceArea() {
        return 2 * (length * width + length * height + width * height);
    }

    @Override
    public double volume() {
        return length * width * height;
    }
}

class WoodBox extends Box {
    private double thickness;

    public WoodBox(double length, double width, double height, double
thickness) {
        super(length, width, height);
        this.thickness = thickness;
    }

    @Override

```

```

        public double wholeSurfaceArea() {
            return super.wholeSurfaceArea() + 4 * (length * thickness + width *
thickness);
        }

        @Override
        public double volume() {
            return super.volume() + 2 * (length * thickness * height + width *
thickness * height);
        }
    }
}

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

        System.out.print("Enter dimensions for Plate (length, width): ");
        double plateLength = scanner.nextDouble();
        double plateWidth = scanner.nextDouble();
        Plate plate = new Plate(plateLength, plateWidth);
        System.out.println("Whole Surface Area of Plate: " +
plate.wholeSurfaceArea());
        System.out.println("Volume of Plate: " + plate.volume());

        System.out.print("\nEnter dimensions for Box (length, width, height):
");
        double boxLength = scanner.nextDouble();
        double boxWidth = scanner.nextDouble();
        double boxHeight = scanner.nextDouble();
        Box box = new Box(boxLength, boxWidth, boxHeight);
        System.out.println("Whole Surface Area of Box: " +
box.wholeSurfaceArea());
        System.out.println("Volume of Box: " + box.volume());

        System.out.print("\nEnter dimensions for Wood Box (length, width,
height, thickness): ");
        double woodBoxLength = scanner.nextDouble();
        double woodBoxWidth = scanner.nextDouble();
        double woodBoxHeight = scanner.nextDouble();
        double woodBoxThickness = scanner.nextDouble();
        WoodBox woodBox = new WoodBox(woodBoxLength, woodBoxWidth,
woodBoxHeight, woodBoxThickness);
        System.out.println("Whole Surface Area of Wood Box: " +
woodBox.wholeSurfaceArea());
        System.out.println("Volume of Wood Box: " + woodBox.volume());

        scanner.close();
    }
}

```

```
}
```

17. Aim of the program : Write a program in Java having three classes Apple, Banana and Cherry. Class Banana and Cherry are inherited from class Apple and each class have their own member function show() . Using Dynamic Method Dispatch concept display all the show() method of each class.

→

```
class Apple {
    public void show() {
        System.out.println("This is Apple.");
    }
}

class Banana extends Apple {
    @Override
    public void show() {
        System.out.println("This is Banana.");
    }
}

class Cherry extends Apple {
    @Override
    public void show() {
        System.out.println("This is Cherry.");
    }
}

public class FruitDemo {
    public static void main(String[] args) {
        Apple apple = new Apple();
        Banana banana = new Banana();
        Cherry cherry = new Cherry();

        apple.show();
        banana.show();
        cherry.show();
    }
}
```

This is Apple.

This is Banana.

This is Cherry.

18. Aim of the program :Write a class Account containing acc_no, balance as data members and two methods as input() for taking input from user and disp() method to display the details. Create a subclass Person which has name an aadhar_no as extra data members and override disp() function. Write the complete program to take and print details of three persons.



```
import java.util.Scanner;

class Account {
    protected int accNo;
    protected double balance;

    public Account(int accNo, double balance) {
        this.accNo = accNo;
        this.balance = balance;
    }

    public void displayDetails() {
        System.out.println("Account Number: " + accNo);
        System.out.println("Balance: " + balance);
    }
}

class Person extends Account {
    private String name;
    private long aadharNo;

    public Person(int accNo, double balance, String name, long aadharNo) {
        super(accNo, balance);
        this.name = name;
        this.aadharNo = aadharNo;
    }

    @Override
    public void displayDetails() {
        super.displayDetails();
        System.out.println("Name: " + name);
        System.out.println("Aadhar Number: " + aadharNo);
    }
}

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

        System.out.println("Enter details for three persons:");
        for (int i = 0; i < 3; i++) {
            System.out.println("\nEnter details for person #" + (i + 1));
```

```

        System.out.print("Account Number: ");
        int accNo = scanner.nextInt();
        System.out.print("Balance: ");
        double balance = scanner.nextDouble();
        scanner.nextLine(); // Consume newline character
        System.out.print("Name: ");
        String name = scanner.nextLine();
        System.out.print("Aadhar Number: ");
        long aadharNo = scanner.nextLong();

        Person person = new Person(accNo, balance, name, aadharNo);
        System.out.println("\nDetails of person #" + (i + 1));
        person.displayDetails();
    }

    scanner.close();
}
}

```

Enter details for three persons:

Enter details for person #1

Account Number: 4029

Balance: 20000

Name: bibek

Aadhar Number: 00004029

Details of person #1

Account Number: 4029

Balance: 20000.0

Name: bibek

Aadhar Number: 4029

Enter details for person #2

Account Number: 4030

Balance: 30000

Name: ram

Aadhar Number: 00004030

Details of person #2

Account Number: 4030

Balance: 30000.0

Name: ram

Aadhar Number: 4030

Enter details for person #3

Account Number: 4031

Balance: 40000

Name: shyam
Aadhar Number: 00004031

Details of person #3
Account Number: 4031
Balance: 40000.0
Name: shyam
Aadhar Number: 4031