

1. Create a class named Person with the following:

- ☐ Instance variables: String name, int age.
- ☐ constructor to initialize the instance variables.
- ☐ An instance method displayInfo() to print the person's name and age.
- ☐ An instance method isAdult() to return true if the person's age is 18 or above.

sol:

```
public class Person {
    String name;
    int age;

    public Person(String name,int age){
        this.name=name;
        this.age=age;
    }
    public void displayInfo() {
        System.out.println("Person Name :"+name);
        System.out.println("Person Age  :"+age);
    }
    public boolean isAdult() {
        return age >=18;
    }
    public static void main(String[] args) {
        Person per=new Person("Mani",22);
        per.displayInfo();

        if(per.isAdult()) {
            System.out.println(per.name+" is Adult ");
        }
        else {
            System.out.println(per.name+" is not Adult");
        }
    }
}
```

2. Define a class called Book with the following properties and methods:

- ☐ Instance variables: String title, String author, double price.
- ☐ Constructors:
 - o A default constructor that initializes the book with default values.
 - o A parameterized constructor that initializes the book with specific values.
- ☐ Method: displayDetails() that prints the book's details.

Instructions:

- ☐ Implement both default and parameterized constructors.
- ☐ Demonstrate object creation using both constructors in a Main class.

sol:

```

public class Book {
    String title;
    String author;
    double price;

    public Book() {

    }

    public Book(String title,String author,double price) {
        this.title=title;
        this.author=author;
        this.price=price;
    }
    public void displayDetails() {
        System.out.println("Book Title  :"+title);
        System.out.println("Book Author  :"+author);
        System.out.println("Book Price   :"+price);
    }
    public static void main(String[] args) {
        Book book1=new Book();
        Book book2=new Book("ZERO TO ONE","PETER THIEL",200d);
        book1.displayDetails();
        book2.displayDetails();
    }
}

```

3.Create a class hierarchy with Vehicle as the base class and Car as the subclass. Implement the following:

❑ Vehicle class:

- o Instance variables: String make, String model.
- o Constructor: Initializes the instance variables.
- o Method: displayInfo() to print vehicle details.

❑ Car class (extends Vehicle):

- o Instance variable: int numberOfDoors.
- o Constructor: Initializes all variables, including those in the Vehicle class using super.
- o Method: Override displayInfo() to include the number of doors.

Instructions:

- ❑ Use the super keyword to access superclass constructors and methods.
- ❑ Demonstrate the inheritance and method overriding in a Main class.

sol:

```

class Vehicle{
    String make;
    String model;

    public Vehicle(String make,String model) {
        this.make=make;
    }
}

```

```

        this.model=model;
    }
    public void displayInfo() {
        System.out.println("Vehicle make :"+make);
        System.out.println("Vehicle model :"+model);
    }
}
class Car extends Vehicle{
    int numberOfDoors;

    public Car(String make,String model,int numberOfDoors) {
        super(make,model);
        this.numberOfDoors=numberOfDoors;
    }

    public void displayInfo() {
        super.displayInfo();
        System.out.println("No.of Doors :"+numberOfDoors);
    }

}

}
public class Hierarchy {
    public static void main(String[] args) {
        Car car=new Car("Bajaj Auto","Pulsar 150",0);
        car.displayInfo();
    }

}

```

4.Design a class Zoo to manage animals in a zoo. Implement the following:

☐ Animal class:

- o Instance variables: String name, int age.
- o Constructor: Initializes instance variables.
- o Method: displayInfo() to print animal details.

☐ Bird class (extends Animal):

- o Instance variable: String species.
- o Constructor: Initializes all variables using super.
- o Method: Override displayInfo() to include the species.

☐ Zoo class:

- o Main method to create instances of Animal and Bird.
- o Demonstrate method overriding, constructor usage, and static methods for counting animals.

Instructions:

☐ Use inheritance, the super keyword, and both static and instance methods.

☐ Demonstrate all concepts in a Main method.

sol:

```

class Zoo {
    static int count = 0;

```

```

        public static void count() {
            count++;
        }

        public static void countOfAnimals() {
            System.out.println("Total Animal: " + count);
        }

public static void main(String a[]){
    Animal animal1 = new Animal("Dog", 3);
    animal1.displayInfo();
    count();
    Bird bird1 = new Bird("Sparrow", 2, "World");
    bird1.displayInfo();
    count();
    countOfAnimals();
}

}

class Animal {
    String name;
    int age;

    Animal(String name, int age) {
        this.name = name;
        this.age = age;
    }

    void displayInfo() {
        System.out.println("Name of the Animal: " + name);
        System.out.println("Age of the Animal: " + age);
    }
}

class Bird extends Animal {
    String species;

    Bird(String name, int age, String species) {
        super(name, age);
        this.species = species;
    }

    void displayInfo() {
        super.displayInfo();
        System.out.println("the Species: " + species);
    }

public static void main(String ar[]){
    Bird bird1 = new Bird("Birds",3,"Eagle");
    bird1.displayInfo();
}

}

```