

**Question 1 and 2 are connected.**

**1. (Inheritance)**

Create a class Number that contains an int attribute and a method printNumber() to display the number. Then, create a subclass PrimeChecker that inherits from Number and adds a method isPrime() to check if the number stored in Number is prime or not. The isPrime() method should return a boolean value indicating if the number is prime. The main is given and output should look like below.

```
public class Main {  
    public static void main(String[] args) {  
        int[] numbers = {2, 15, 29, 8, 13};  
  
        for (int num : numbers) {  
            PrimeChecker primeChecker = new PrimeChecker(num);  
            primeChecker.printNumber();  
            if (primeChecker.isPrime()) {  
                System.out.println(num + " is a prime number.\n");  
            } else {  
                System.out.println(num + " is not a prime number.\n");  
            }  
        }  
    }  
}
```

```
The number is: 2  
2 is a prime number.  
  
The number is: 15  
15 is not a prime number.  
  
The number is: 29  
29 is a prime number.  
  
The number is: 8  
8 is not a prime number.  
  
The number is: 13  
13 is a prime number.
```

**2. (Hierarchical Inheritance)**

Extend the PrimeChecker class to create a class PrimeFactorizer that adds a method printFactors() to print all prime factors of the number. Override the isPrime() method to check if the number has any prime factors. For example, number 15 has the factors as 3 and 5. The main is given and output should look like below.

```
public class Main {  
    public static void main(String[] args) {  
        int[] numbers = {2, 15, 29, 8, 13};  
  
        for (int num : numbers) {  
            PrimeFactorizer primeFactorizer = new PrimeFactorizer(num);  
            primeFactorizer.printNumber();  
            if (primeFactorizer.isPrime()) {  
                System.out.println(num + " is a prime number.\n");  
            } else {  
                System.out.println(num + " is not a prime number.");  
                primeFactorizer.printFactors();  
                System.out.println();  
            }  
        }  
    }  
}
```

```
The number is: 2  
2 is a prime number.  
  
The number is: 15  
15 is not a prime number.  
Prime factors of 15:  
3 5  
  
The number is: 29  
29 is a prime number.  
  
The number is: 8  
8 is not a prime number.  
Prime factors of 8:  
2 2 2  
  
The number is: 13  
13 is a prime number.
```

**Question 3 and 4 are connected.**

**3. (Hierarchical Inheritance)**

Create a base class called Document with attributes for title and content. Implement a method printSummary() that prints a brief summary of the document. Create two subclasses: TextDocument and PdfDocument. The TextDocument class should override the printSummary() method to print the first 100 characters of the content, while the PdfDocument class should override it to print the first line of the content.

The main is given and output should look like below.

```
public class Main {
    public static void main(String[] args) {
        Document textDoc = new TextDocument("Text Doc 1", "This is a sample text document content that should be summarized.");
        Document pdfDoc = new PdfDocument("PDF Doc 1", "Line 1: This is the first line.\nLine 2: This is the second line.");

        textDoc.printSummary();
        System.out.println();
        pdfDoc.printSummary();
    }
}
```

```
Text Document Title: Text Doc 1
Content Summary: This is a sample text document content that should be summarized.

PDF Document Title: PDF Doc 1
Content Summary: Line 1: This is the first line.
```

**4. (Inheritance)**

Enhance the Document class to include a method getDocumentType() that returns a string representing the type of document. Override this method in TextDocument to return "Text Document" and in PdfDocument to return "PDF Document".

The main is given and output should look like below.

```
public class Main {
    public static void main(String[] args) {
        Document textDoc = new TextDocument("Text Doc 1", "This is a sample text document content that should be summarized.");
        Document pdfDoc = new PdfDocument("PDF Doc 1", "Line 1: This is the first line.\nLine 2: This is the second line.");

        System.out.println(textDoc.getDocumentType());
        textDoc.printSummary();
        System.out.println();

        System.out.println(pdfDoc.getDocumentType());
        pdfDoc.printSummary();
    }
}
```

```
Text Document
Text Document Title: Text Doc 1
Content Summary: This is a sample text document content that should be summarized.

PDF Document
PDF Document Title: PDF Doc 1
Content Summary: Line 1: This is the first line.
```

## 5. (Inheritance)

Create a base class named Vehicle with two attributes: type (String) initialized to "4W" and maxSpeed (int) initialized to 100. Include a parameterized constructor to initialize these attributes. Next, create a subclass of Vehicle called Car. This class should add an additional attribute trans (String) for the transmission type and implement a parameterized constructor to initialize trans. The constructor of Car should call the base class constructor with "4W" and 150 as arguments. The main is given and output should look like below.

```
public class Main {  
    public static void main(String[] args) {  
        Car c = new Car("Auto");  
        System.out.println(c.type + " " + c.maxSpeed + " " + c.trans);  
    }  
}
```

```
4W 150 Auto
```

## 6. (Hierarchical Inheritance)

Create a class Person with a method speak() that displays a message "The person is speaking". Create two subclasses "Student" and "Teacher" that extend "Person" and implement the speak() method to display "The student is asking a question" and "The teacher is giving a lecture" respectively. The main is given and output should look like below.

```
public class Main {  
    public static void main(String[] args) {  
        Person person = new Person();  
        Student student = new Student();  
        Teacher teacher = new Teacher();  
  
        person.speak();  
        student.speak();  
        teacher.speak();  
    }  
}
```

```
The person is speaking  
The student is asking a question  
The teacher is giving a lecture
```

## 7. (Inheritance)

Create a class called Airplane with attributes for flight number, destination, and departure time. Include methods to check the flight status and delay. Additionally, create a subclass called InternationalFlight that inherits from Airplane and adds an attribute for the country of origin. Override the flight status method in InternationalFlight to include information about international flights.

```
public class Main {  
    public static void main(String[] args) {  
        Airplane domesticFlight = new Airplane("FL123", "New York", "10:00 AM");  
        domesticFlight.checkFlightStatus();  
        domesticFlight.checkDelay();  
  
        InternationalFlight internationalFlight = new InternationalFlight("FL456", "London", "2:00 PM", "France");  
        internationalFlight.checkFlightStatus();  
        internationalFlight.checkDelay();  
    }  
}
```

```
Flight FL123 to New York is on time.  
Flight FL123 has no delays.  
International Flight FL456 to London from France is on time.  
Flight FL456 has no delays.
```

## 8. (Linked list)

Create a class named `LinkedList` that holds an `int[]` array (data) and initializes it via a constructor. Implement a display method in this class to print all node data. Also, define a `passMessage` method to send and print a message for each node. The main is given and output should look like below.

```
public class Main {
    public static void main(String[] args) {
        int[] values = {1, 2, 3};
        LinkedList list = new LinkedList(values);

        System.out.println("Displaying nodes:");
        list.display();

        System.out.println("\nPassing message through nodes:");
        list.passMessage("Hello from the list!");
    }
}
```

```
Displaying nodes:
Node data: 1
Node data: 2
Node data: 3

Passing message through nodes:
Message at node with data 1: Hello from the list!
Message at node with data 2: Hello from the list!
Message at node with data 3: Hello from the list!
```

## 9. (Pointers)

You will write a `twoSum` method inside the `TwoSum` class. This method takes an integer array (input) and a target value (`targetValue`). It should check if there are two numbers in the array that sum up to the target value. If such a pair exists, return `true`; otherwise, return `false`. Assume the array is sorted. (This method is designed to work with sorted arrays.) For example, with the array `[1, 3, 5, 7, 9]`, you cannot obtain the number 20 by summing any two numbers, so the result should be `false`. However, if the target value were 8, since you can obtain it by summing 7 and 1, the result would be `true`. The main is given and output should look like below.

```
public class Main {
    public static void main(String[] args) {
        TwoSum ts = new TwoSum();

        int[] inputArray = {1, 2, 3, 4, 5, 6};
        int target = 10;

        boolean result = ts.twoSum(inputArray, target);
        System.out.println("Is there a pair with the target sum? " + result);

        int[] anotherArray = {1, 3, 5, 7, 9};
        int anotherTarget = 20;
        boolean anotherResult = ts.twoSum(anotherArray, anotherTarget);
        System.out.println("Is there a pair with the target sum? " + anotherResult);
    }
}
```

```
Is there a pair with the target sum? true
Is there a pair with the target sum? false
```

```
public class Main {
    public static void main(String[] args) {
        Vehicle car = new Car();
        car.drive();

        Vehicle motorcycle = new Motorcycle();
        motorcycle.drive();

        Vehicle electricCar = new ElectricCar();
        electricCar.drive();
    }
}
```

```
Driving a car
Riding a motorcycle
Driving an electric car
```

#### 10. (Inheritance and constructor)

Create a superclass named Vehicle with two attributes: model (String) and year (int), and include a constructor. Then, create a subclass named Car that extends Vehicle and has an additional attribute numberOfDoors (int). In the Car class, write a constructor that calls the Vehicle class constructor. In the Main class, create an instance of the Car class, set the model, year, and number of doors, and print these details to the screen. The main is given and output should look like below.

```
public class Main {  
    public static void main(String[] args) {  
        Car myCar = new Car("Toyota Corolla", 2020, 4);  
        myCar.displayInfo();  
    }  
}
```

Model: Toyota Corolla  
Year: 2020  
Number of Doors: 4

#### 11. (Inheritance, overriding and constructor)

Create a superclass named Employee with two attributes: name (String) and salary (double), and define a method called calculateBonus that returns 0 to calculate the employee's bonus. Then, create a subclass named Manager that extends Employee and overrides the calculateBonus method to compute the manager's bonus as 10% of the salary. Also, create another subclass named Intern that extends Employee and overrides the calculateBonus method to return 5% of the salary for the intern's bonus. In the Main class, create instances of Manager and Intern, calculate their bonuses, and print the results to the screen. The main is given and output should look like below.

```
public class Main {  
    public static void main(String[] args) {  
        Manager manager = new Manager("Alice", 50000);  
        Intern intern = new Intern("Bob", 20000);  
  
        System.out.println(manager.name + "'s bonus: $" + manager.calculateBonus());  
        System.out.println(intern.name + "'s bonus: $" + intern.calculateBonus());  
    }  
}
```

Alice's bonus: \$5000.0  
Bob's bonus: \$1000.0

#### 12. (Overloading)

You are required to create a Calculator class with the following methods: add(int a, int b) which adds two integers and returns the result; add(double a, double b) which adds two double values and returns the result; multiply(double a, int b) which multiplies a double value by an integer and returns the result. All methods will have 2 attributes as a and b. The main is given and output should look like below.

```
public class Main {  
    public static void main(String[] args) {  
        MathOperations mo = new MathOperations();  
  
        System.out.println("Sum of 5 and 10: " + mo.add(5, 10));  
        System.out.println("Sum of 5.5 and 10.5: " + mo.add(5.5, 10.5));  
        System.out.println("Multiplication of 5.5 and 4: " + mo.multiply(5.5, 4));  
    }  
}
```

Sum of 5 and 10: 15  
Sum of 5.5 and 10.5: 16.0  
Multiplication of 5.5 and 4: 22.0

### 13. (Overloading)

Create a class named SimpleConverter with two methods: convert(double value), which converts the given value from meters to feet, and convert(double value, String unit), which converts the given value to centimeters based on the specified unit. When the unit is "meters," the method should convert the value to feet; when the unit is "inches," it should convert the value to centimeters. (Hint: Overloading, meters => feet is x3.28084, inches => cm is x2.54) The main is given and output should look like below.

```
public class Main {
    public static void main(String[] args) {
        SimpleConverter converter = new SimpleConverter();

        System.out.println("Convert 10 meters to feet: " + converter.convert(10)); // metre -> feet
        System.out.println("Convert 25 inches to cm: " + converter.convert(25, "inches")); // inch -> cm
    }
}
```

```
Convert 10 meters to feet: 32.8084
Convert 25 inches to cm: 63.5
```

Question 16 and 17 are connected.

### 14. (Array)

Create a class named ArrayManipulator. This class should include a method called sumArray that calculates and returns the sum of all elements in the given integer array with an int[] numbers attribute. The main is given and output should look like below.

```
public class Main {
    public static void main(String[] args) {
        ArrayManipulator manipulator = new ArrayManipulator();

        int[] numbers = {4, 7, 1, 8, 5};

        System.out.println("Maximum value: " + manipulator.findMax(numbers));
    }
}
```

```
Maximum value: 8
```

### 15. (Array)

Create a class named `ArrayManipulator` with an `int[] numbers` attribute. Add a method named `sumArray(int[] numbers)` to this class. This method should calculate and return the sum of all elements in the provided integer array. The main is given and output should look like below.

```
public class Main {
    public static void main(String[] args) {
        ArrayManipulator manipulator = new ArrayManipulator();

        int[] numbers = {4, 7, 1, 8, 5};

        System.out.println("Sum of array elements: " + manipulator.sumArray(numbers));
    }
}
```

```
Sum of array elements: 25
```

## 16. (Constructor)

Write a class to check whether a number is a Duck Number or not. A Duck number is a number which has zeroes present in it, but there should be no zero present in the beginning of the number. For example 3210, 7056, 8430709 are all duck numbers whereas 08237, 04309 are not. isDuckNumber() method should have number attribute as type of String. The main is given and output should look like below.

```
import java.util.Arrays;
public class Main {

    public static void main(String[] args) {
        String[] testNumbers = {"3210", "7056", "8430709", "08237", "04309"};

        for (String number : testNumbers) {
            if (DuckNumberChecker.isDuckNumber(number)) {
                System.out.println(number + " is a Duck Number.");
            } else {
                System.out.println(number + " is not a Duck Number.");
            }
        }
    }
}
```

```
3210 is a Duck Number.
7056 is a Duck Number.
8430709 is a Duck Number.
08237 is not a Duck Number.
04309 is not a Duck Number.
```

## 17. (ArrayList)

Write a Java program to replace the second element of an ArrayList with the specified element. You should write replaceSecondElement() method with ArrayList<String> list and String newElement attributes. In addition, please make sure to add a printList method as well for the printing operation. The main is given and output should look like below.

```
public class Main {
    public static void main(String[] args) {
        ArrayList<String> list = new ArrayList<>();
        list.add("Apple");
        list.add("Banana");
        list.add("Cherry");
        list.add("Date");

        System.out.println("Original ArrayList:");
        ReplaceElement.printList(list);

        String newElement = "Blueberry";

        ReplaceElement.replaceSecondElement(list, newElement);

        System.out.println("\nArrayList after replacing the second element:");
        ReplaceElement.printList(list);
    }
}
```

```
ArrayList after replacing the second element:
Apple
Blueberry
Cherry
Date
```