

## CDAC Mumbai

### Lab Assignment: Flowchart and Java Programming

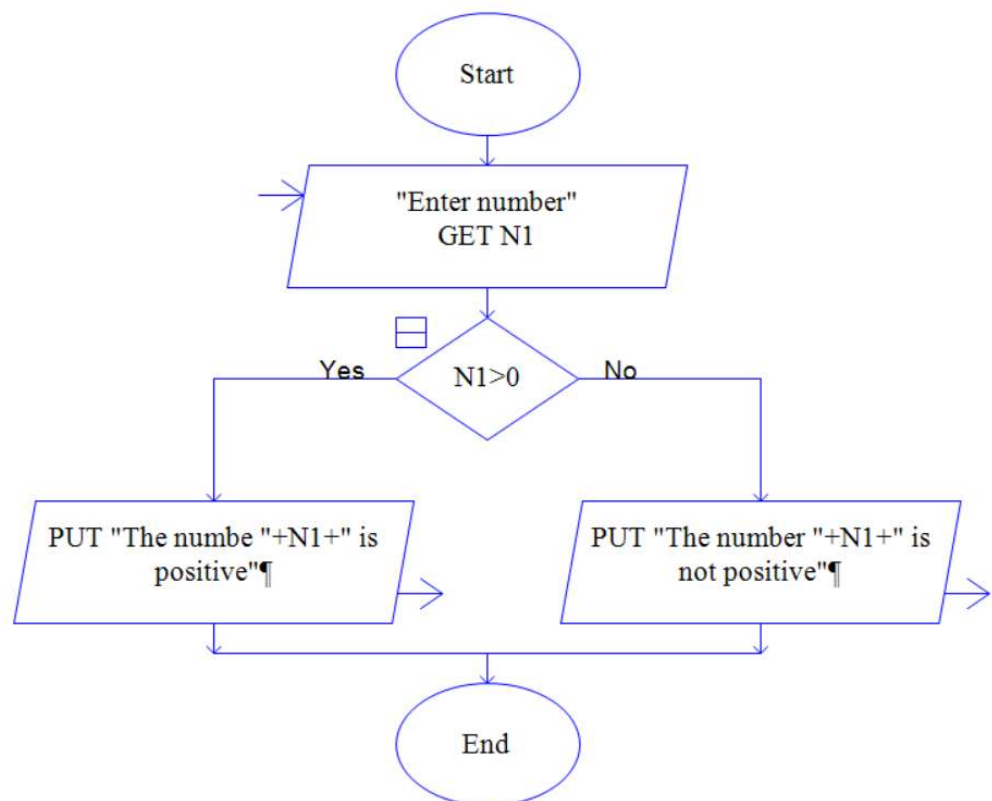
#### Instructions:

1. For each of the following questions, first **create a flowchart** to outline the logic.
2. After completing the flowchart, **write a Java program** to implement the logic based on your flowchart.
3. Ensure your code follows basic Java syntax and logic.
4. You can explore user input (NOT MANDATORY)

#### Flowchart + Java Program Questions

##### 1. Check Positive Number:

- **Task:** Create a flowchart to check whether a number is positive.



- **Next Step:** Write a Java program that checks if a predefined number is positive using an if-else statement and prints the appropriate message.

```
import java.util.*;
```

```
public class Main{
```

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

```

Scanner sc =new Scanner (System.in);

int number = sc.nextInt();

if (number > 0) {

    System.out.println("The number " + number + " is positive.");

} else {

    System.out.println("The number " + number + " is non positive.");

}

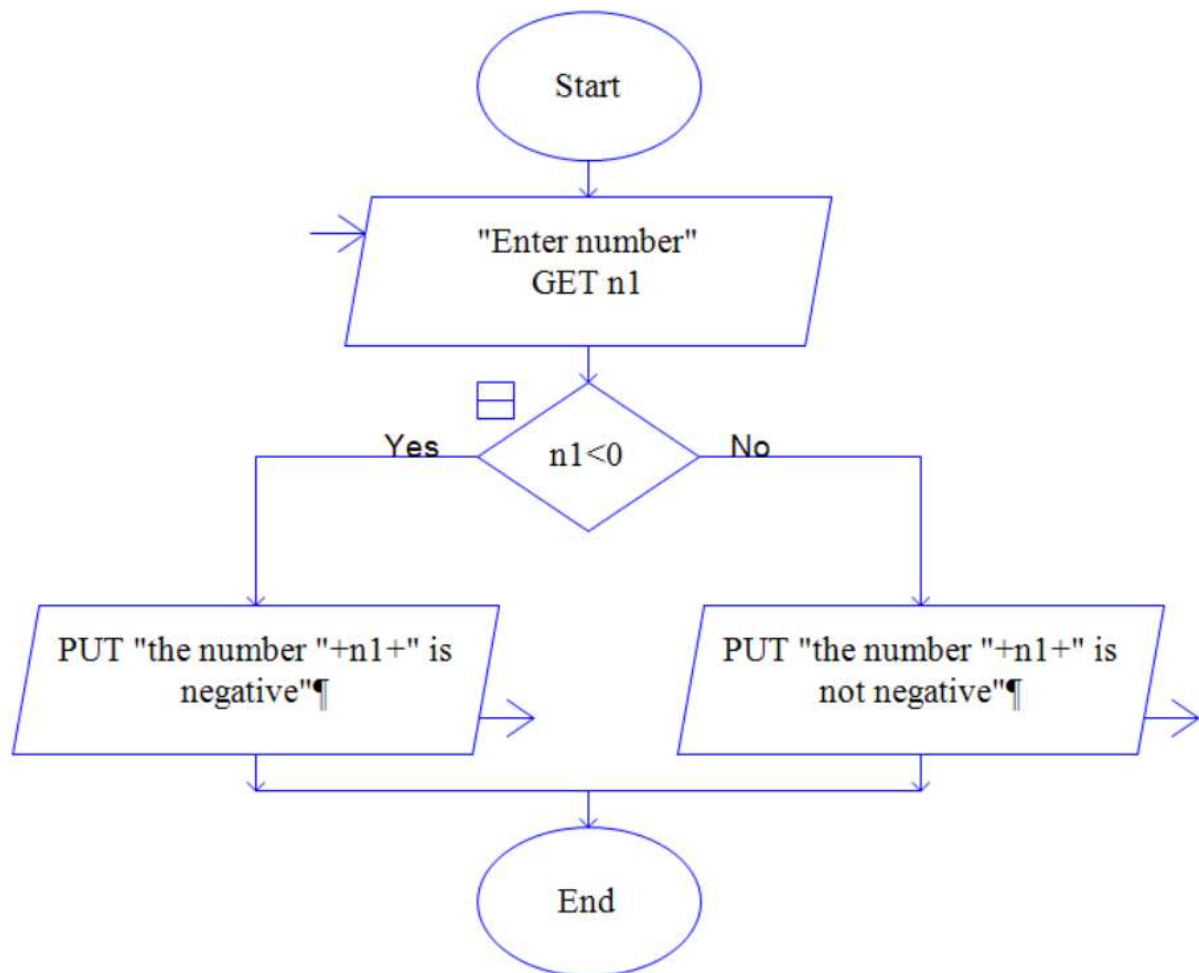
}

}

```

## 2. Check Negative Number:

- **Task:** Create a flowchart to check whether a number is negative.



- **Next Step:** Write a Java program that checks if a predefined number is negative using an if-else statement and displays the result.

```
import java.util.*;

public class Main{

    public static void main(String[] args) {

        Scanner sc =new Scanner (System.in);

        int number = sc.nextInt();

        if (number< 0) {

            System.out.println("The number " + number + " is negative.");

        } else {

            System.out.println("The number " + number + " is non negative.");

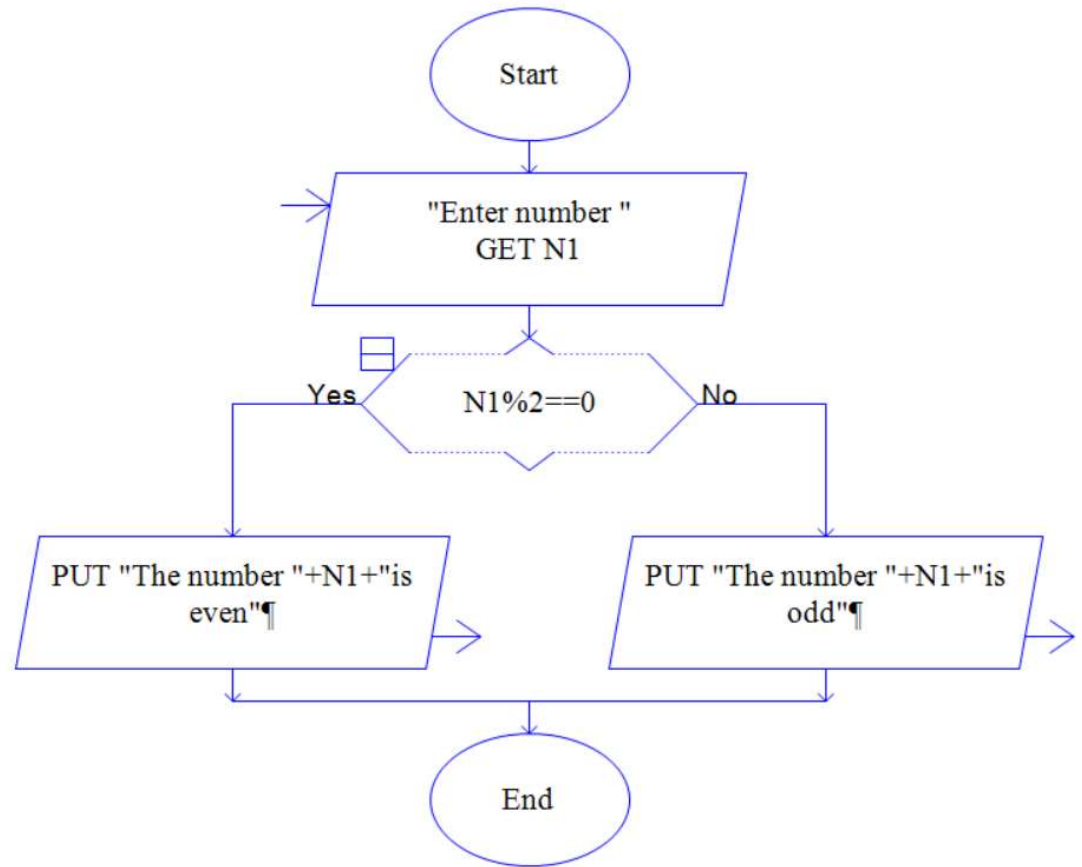
        }

    }

}
```

### 3. Check Odd or Even Number:

- **Task:** Create a flowchart to determine whether a number is odd or even.



- **Next Step:** Write a Java program that checks if a predefined number is odd or even. Use an if-else statement and the modulus operator (%) to determine whether the number is divisible by 2 or not.

```
import java.util.*;
```

```
public class Main{
```

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

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

```
        int number = sc.nextInt();
```

```
        if (number % 2 == 0) {
```

```
            System.out.println("The number " + number + " is even.");
```

```
        } else {
```

```
            System.out.println("The number " + number + " is odd.");
```

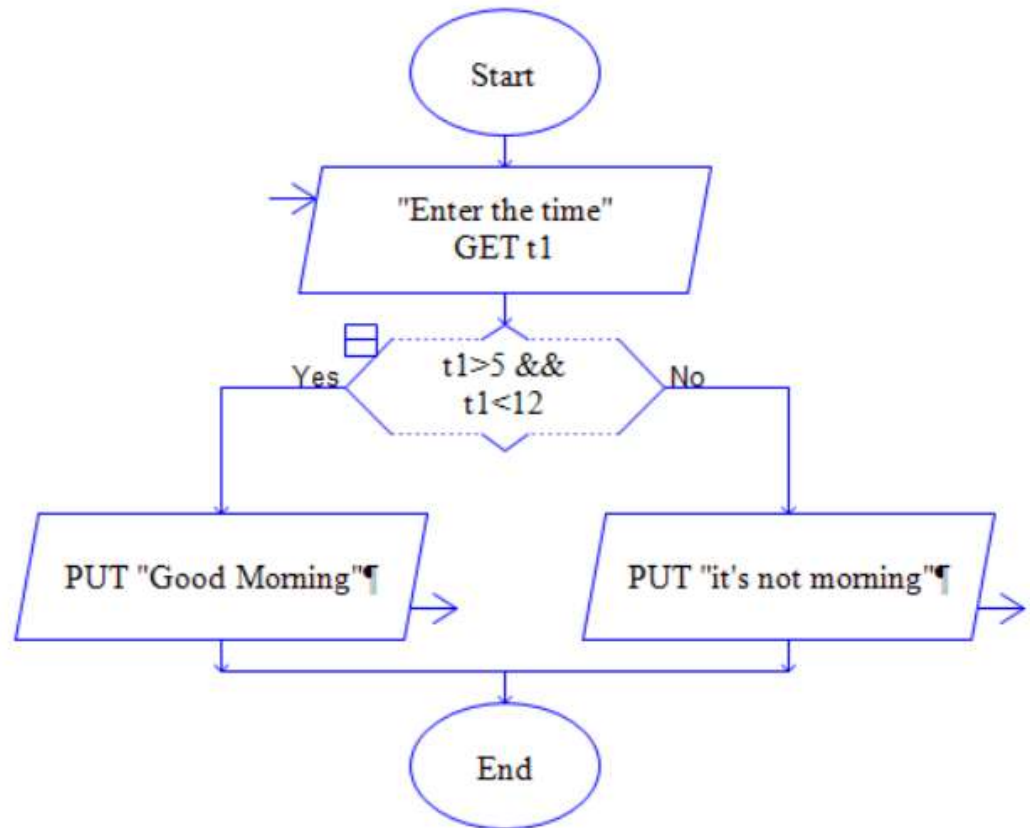
```
        }
```

```
    }
```

}

#### 4. Display Good Morning Message Based on Time:

- **Task:** Create a flowchart to display a "Good Morning" message based on a given time.



- **Next Step:** Write a Java program that displays a "Good Morning" message if the predefined time is between 5 AM and 12 PM. Use an if statement to implement the logic.

```
public class Main {  
  
    public static void main(String[] args) {  
  
        Scanner sc = new Scanner(System.in);  
  
        int hour = sc.nextInt();  
  
        if (hour >= 5 && hour < 12) {  
  
            System.out.println("Good Morning");  

```

```

    } else {

        System.out.println("It's not morning");

    }

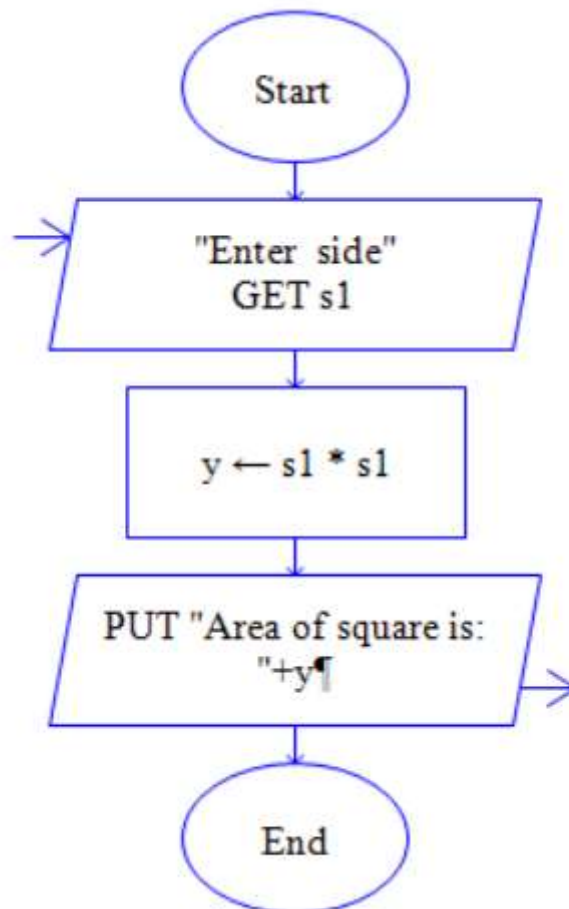
}

}

```

##### 5. Print Area of a Square:

- **Task:** Create a flowchart to calculate and print the area of a square.

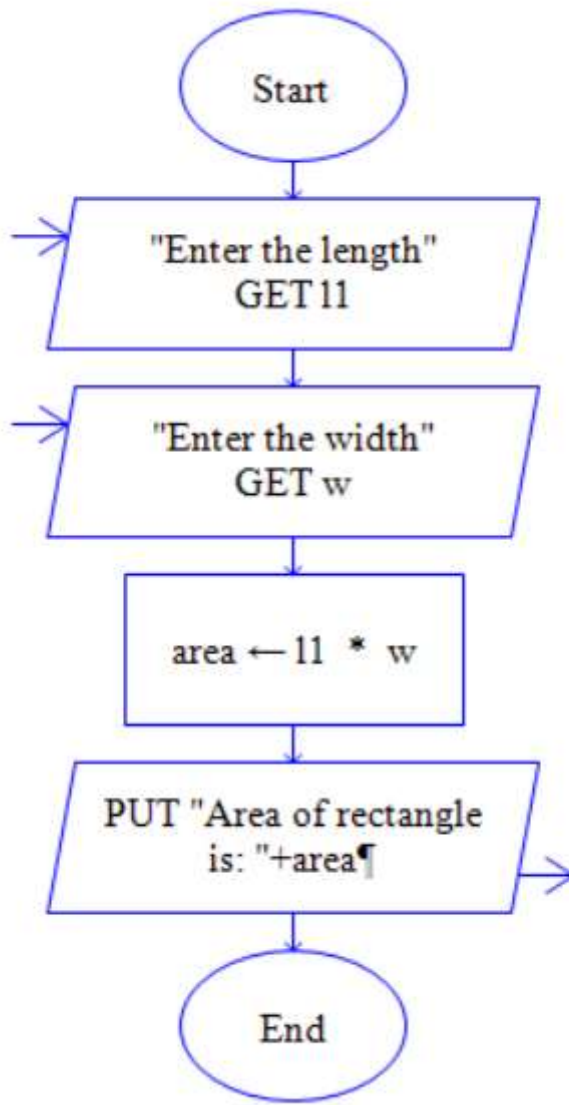


- **Next Step:** Write a Java program that calculates the area of a square using the formula  $\text{area} = \text{side} * \text{side}$ . Use a predefined side length.

```
public class Main{  
    public static void main(String[] args) {  
        scanner sc =new Scanner(System.in);  
        double side = sc.nextDouble();  
        double area = side * side;  
  
        System.out.println("The area of the square is: " + area);  
    }  
}
```

#### 6. **Print Area of a Rectangle:**

- **Task:** Create a flowchart to calculate and print the area of a rectangle.



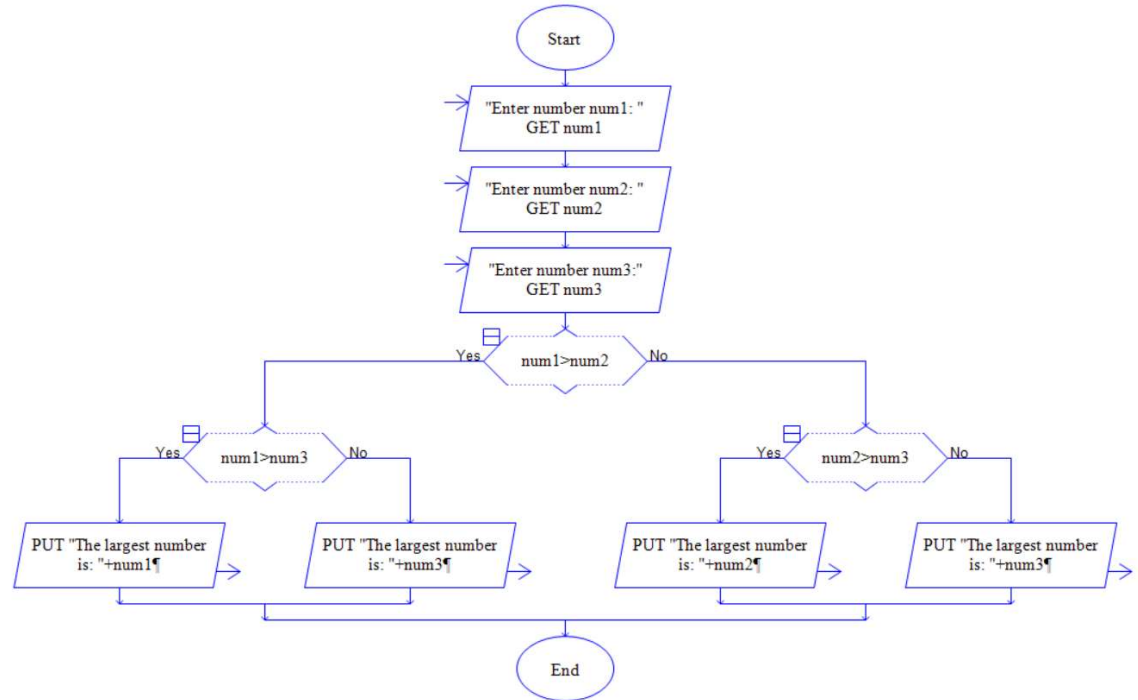
- **Next Step:** Write a Java program that calculates the area of a rectangle using the formula  $area = length * width$ . Use predefined values for length and width.

```
public class Main{  
    public static void main(String[] args) {  
        double length = 12.5;  
        double width = 8.0;  
        double area = length * width;  
        System.out.println("The area of the rectangle is: " + area);  
    }  
}
```



## 7. Find the Largest of Three Numbers:

- **Task:** Create a flowchart to find the largest of three numbers.



- **Next Step:** Write a Java program that finds and prints the largest of three predefined numbers using if-else statements.

```
public class Main{  
  
    public static void main(String[] args) {  
  
        int num1 = 10;  
  
        int num2 = 20;  
  
        int num3 = 15;  
  
        int largest;  
  
        if (num1 >= num2 && num1 >= num3) {  
  
            largest = num1;  
  
        } else if (num2 >= num1 && num2 >= num3) {  
  
            largest = num2;  
  
        } else {  
  
            largest = num3;  
  
        }  
    }  
}
```

```
    }  
  
    System.out.println("The largest number is: " + largest);  
  
    }  
  
}
```

---

### **Food for Thought: Research and Read More About**

- **History of Java:** Explore the origin and development of the Java programming language. Who created Java, and why was it developed? How has it evolved over time?
- **How Java is Useful & Problems It Solves:** Research the specific problems Java addresses in software development. Why is Java preferred for certain types of projects (e.g., web development, mobile apps, enterprise systems)? What are some of its key strengths?
- **Role of the Java Virtual Machine (JVM):** Investigate the purpose of the JVM in the execution of Java programs. How does it enable Java's platform independence (i.e., "Write Once, Run Anywhere")?
- **Java Runtime Environment (JRE):** Read about how the JRE fits into the picture when running Java applications. What does the JRE provide, and why is it essential?
- **Difference Between JDK, JRE, and JVM:** Understand the differences and relationships between the **Java Development Kit (JDK)**, **Java Runtime Environment (JRE)**, and **Java Virtual Machine (JVM)**. How do these components work together when a Java program is written, compiled, and executed?
- **Memory Areas in JVM:** Explore the different types of memory areas within the JVM, such as the **Heap**, **Stack**, and **Method Area**. What roles do these memory areas play during the execution of a Java program?
- **Primitive Data Types in Java:** Learn about Java's primitive data types. What are they, and how are they different from reference data types? List and explain the eight primitive data types in Java.