Control Statements in Java | Control Flow in Java

Java compiler executes the code from top to bottom. The statements in the code are executed according to the order in which they appear. However, Java provides statements that can be used to control the flow of Java code. Such statements are called control flow statements. It is one of the fundamental features of Java, which provides a smooth flow of program.

Java provides three types of control flow statements.

1. **Decision Making statements**
   - **if statements.**
   - **switch statement.**
2. **Loop statements**
   - **do while loop.**
   - **while loop.**
   - **for loop.**
   - **for-each loop.**
3. **Jump statements**
   - **break statement.**
   - **continue statement.**

1. Decision-Making statements

As the name suggests, decision-making statements decide which statement to execute and when. Decision-making statements evaluate the Boolean expression and control the program flow depending upon the result of the condition provided. There are two types of decision-making statements in Java, i.e., If statement and switch statement.

➢ If Statement

In Java, the "if" statement is used to evaluate a condition. The control of the program is diverted depending upon the specific condition. The condition of the If statement gives a Boolean value, either true or false. In Java, there are four types of if-statements given below.

- ❖ **Simple if statement.**
- ❖ **if-else statement.**
- ❖ **if-else-if ladder.**
- ❖ **Nested if-statement.**

Simple if statement

- It is the most basic statement among all control flow statements in Java. It evaluates a Boolean expression and enables the program to enter a block of code if the expression evaluates to true.
- Syntax of if statement is given below.

Syntax

```
if (condition) {
statement 1;          //executes when condition is true
}
```

Example # 1:

Write a program in Java to read two integer numbers, then find the largest number between them?

The code is:

```
public class LargestNumber {
    public static void main(String[] args) {
        int a = 20;
        int b = 10;
        if (a > b){
            System.out.println("The Lagest Number is " + a);

        }
    }
}
```

if-else statement

The if-else statement is an extension to the if-statement, which uses another block of code, i.e., else block. The else block is executed if the condition of the if-block is evaluated as false.

Syntax

if (condition) {

statement 1; //executes when condition is true

}

else {

statement 2; //executes when condition is false

}

Example # 2:

Write a program in Java to read two integer numbers, then find the largest number between them?

The code is:

```
public class LargestNumber {
    public static void main(String[] args) {
        int a = 20;
        int b = 10;
        if (a > b){
            System.out.println("The Lagest Number is " + a);
        } else {
            System.out.println("The Lagest Number is " + b);
```

```
            }
        }
}
```

if-else-if ladder

The if-else-if statement contains the if-statement followed by multiple else-if statements. In other words, we can say that it is the chain of if-else statements that create a decision tree where the program may enter in the block of code where the condition is true. We can also define an else statement at the end of the chain.

Syntax

```
if (condition 1) {
statement 1;}        //executes when condition 1 is true
else if (condition 2) {
statement 2;}        //executes when condition 2 is true
else {statement;} //executes when all the conditions are false
```

Example # 3:

Write a program in Java to read two integer numbers, then find the largest number between them?

The code is:

```
public class LargestNumber {
    public static void main(String[] args) {
        int a = 10;
        int b = 20;
```

```
        if (a > b){
            System.out.println("The Lagest Number is " + a);
        }
        else if (b > a) {
            System.out.println("The Lagest Number is " + b);
        }
        else {
            System.out.println("Two numbers are equal");
        }
    }
}
```

Nested if-statement

In nested if-statements, the if statement can contain if or if-else statement inside another if or else-if statement.

Syntax

```
if (condition 1) {
statement 1;        //executes when condition 1 is true
if (condition 2) {
statement 2;        //executes when condition 2 is true
}
}
else {
```

statement;

} //executes when all the conditions are false

Example # 4:

Write a program in Java to read three integer numbers, then find the largest number between them?

The code is:

```
public class LargestNumber {
    public static void main(String[] args) {
        int a = 10;
        int b = 20;
        int c = 30;
        if (a >= b) {
            if (a >= c)
                System.out.println("The Lagest Number is " + a);
            else
                System.out.printjln("The Lagest Number is " + c);
        }
        else
        {
            if (b >= c)
                System.out.println("The Lagest Number is " + b);
            else
                System.out.println("The Lagest Number is " + c);
        }}}
```

Example # 5:

Write a program in Java to read two integer numbers, then find the smallest number between them?

The code is:

```
public class SmallestNumber {
    public static void main(String[] args) {
        int x = 20;
        int y = 10;
        if (x == y)
        System.out.println("The Two Numbers are Equal ");
        else if (x < y) {
        System.out.println("The Smallest Numbers is "+ x); }
        else
        System.out.println("The Smallest Numbers is " + y);
        }
    }
```

Example # 6:

Write a program in Java to read two integer numbers, then find the smallest number between them?

The code is:

```
import java.util.Scanner;
public class SmallestNumber2 {
    public static void main(String[] args) {
        int num1, num2;
        Scanner input = new Scanner (System.in);
        System.out.println("Enter The First Number");
        num1 = input.nextInt();
        System.out.println("Enter The Second Number");
        num2 = input.nextInt();
        if (num1 == num2)
           System.out.println("The Two Numbers are Equal ");
        else if (num1 < num2) {
     System.out.println("The Smallest Numbers is "+ num1); }
        else
      System.out.println("The Smallest Numbers is " + num2);
    }
}
```

Example # 7:

Write a program in Java to read three integer numbers, then find the largest number between them?

The code is:

```
import java.util.Scanner;
public class LargestNumber2 {
    public static void main(String[] args) {
        int a, b, c;
        Scanner input = new Scanner (System.in);
        System.out.println("Enter The First Number");
        a = input.nextInt();
        System.out.println("Enter The Second Number");
        b = input.nextInt();
        System.out.println("Enter The Third Number");
        c = input.nextInt();
        if (a >= b && a >= c) {
            System.out.println("The Lagest Number is " + a);
            else if
          System.out.println("The Lagest Number is " + c); }
        else {
            if (b >= c)
            System.out.println("The Lagest Number is " + b);
            else
         System.out.println("The Lagest Number is " + c);}}}
```

Example # 8:

Write a program in Java to read three integer numbers, then find the largest number between them?

The code is:

```
import java.util.Scanner;
public class LargestNumber3 {
    public static void main(String[] args) {
        int a, b, c;
        Scanner input = new Scanner (System.in);
        System.out.println("Enter The First Number");
        a = input.nextInt();
        System.out.println("Enter The Second Number");
        b = input.nextInt();
        System.out.println("Enter The Third Number");
        c = input.nextInt();
        if (a >= b && a >= c)
            System.out.println("The Lagest Number is " + a);
            else if (b >= a && b >= c)
            System.out.println("The Lagest Number is " + b);
        else
            System.out.println("The Lagest Number is " + c);
     }
}
```

Homework:

1. Write a program in Java to read four numbers from the user, then find the summation between them?
2. Write a program in Java to read four numbers from the user, then find the largest number between them?
3. Write a program in Java to calculate the value of Y from the following equation?
   $Y = 1/X^2 - 81$.
4. Write a program to print the student average as follows:

   average >100 and average < 0 print "The average is out of range"

   average >=0 and average < 50 print "The average is Fail"

   average >=50 and average <= 59 print "The average is Accepted"

   average >=60 and average <= 69 print "The average is Medium"

   average >=70 and average <= 79 print "The average is Good"

   average >=80 and average <= 89 print "The average is Very Good"

   average >=90 and average <= 100 print "The average is Excellent"

Thank You

د. محمد رضا الكعبي

Mobile: 07801582655

E-mail: mridha.hamoodi@uokufa.edu.iq

" اعلم بأن اجمل العطور ليس ما تضعه على جسدك وملابسك،

بل ما تضعه على لسانك، ويشعر به الآخرون،

عطر الكلمة الطيبة "