Java Decision Making

Java Decision Making

In Java programming language, Decision Making can be done with the help of following statements:

- if
- if else
- if else if
- switch

Generally during the process of decision making, there would be a condition or state of situation on which your course of action depends.

Now, let us see, each of these decision making statements in detail.

Java If Statement

If a condition is satisfied, then pick a course of action, else continue with the subsequent statements. For an example, if(you wake up before 11am), then go for the breakfast in fridge.

Example 1 – Java If

IfExample.java

```
public class IfExample {
    public static void main(String[] args) {
        int time_hours = 10;
        if(time_hours<=11){
            System.out.println("Eat breakfast which is present in the fridge.");
        }
        System.out.println("Execution continues after if-else block");
    }
}
```

Output

```
Eat breakfast which is present in the fridge.
Execution continues after if-else block
```

Java If Else Statement

If a condition is satisfied, then pick a course of action, else pick another course of action. For an example, if(you wake up before 11am), then go for the breakfast in fridge. else trash it.

Example 2 – Java If Else

IfElseExample.java

```
public class IfElseExample {
    public static void main(String[] args) {
        int time_hours = 10;
        if(time_hours<=11){
            System.out.println("Eat breakfast which is present in the fridge.");
        } else{
            System.out.println("Trash it");
        }
        System.out.println("Execution continues after if-else block");
    }
}
```

Output

```
Eat breakfast which is present in the fridge.
Execution continues after if-else block
```

You can change the value of variable “time_hours” and play around with the code for better understanding of the execution flow.

Java If Else If Statement

If a condition is satisfied, then pick a course of action, else check if another condition in which you are interested is satisfied, to take another course of action, (this process of could go on for ever if you like) and when nothing works out(no condition you are expecting in the if blocks gets satisfied) you always have the else option waiting for you at the end. Make good use of it.

Example 3 – Java If Else If

IfElseIfExample.java

```
public class IfElseIfExample {
    public static void main(String[] args) {
        int time_hours = 10;
        if(time_hours<=11){ System.out.println("Eat breakfast which is present in the fr
            System.out.println("Eat lunch from outside.");
        } else{
```

```
            System.out.println("Go to sleep.");
        }
        System.out.println("Execution continues after if-else block");
    }
}
```

Output

```
Eat lunch from outside.
Execution continues after if-else block
```

You can change the value of variable “time_hours” and play around with the code for better understanding of the execution flow.

Java Switch Case Statement

Switch actually depends on the state of a single variable or state of an expression. For instance, if you consider the day of week as a variable, in **case** of Monday, curse yourself and go to work, in **case** of Tuesday, curse one more time and go to work, in **case** of Saturday, give yourself some credit for bearing the previous five days and treat yourself the best.

Example 4 – Java Switch Case

SwitchCaseExample.java

```
public class SwitchCaseExample {
    public static void main(String[] args) {
        int time_hours = 12;
        switch(time_hours){
            case(11):{
                System.out.println("Eat breakfast which is present in the fridge.");
                break;
            }
            case(12):{
                System.out.println("Eat lunch from outside.");
                break;
            }
            default:{
                System.out.println("Go to sleep.");
            }
        }
        System.out.println("Execution continues after if-else block");
    }
}
```

Output

```
Eat lunch from outside.
Execution continues after if-else block
```

If any of the case doesn't match, **default** block gets executed. **break;** statement should be given as the last statement of each **case** block. If not, the default would execute after the execution of **case** block.

In Java Decision making, out of all the conditions or cases, there could be only one condition that could be satisfied, and when the condition is satisfied, the course of action intended for that condition is only executed.

Conclusion

In this Java Tutorial, we have learnt about Decision Making Statements in Java Programming Language with example programs. In our next tutorial, we shall learn Loop Statements in Java.

✦ Java Math