

Conditional Control Statements

Conditional Control Statements allows the program to select between the alternatives during the program execution. They are also called as decision-making statements or selection statements.

- **If-Else statement**

If the condition is true then, it will execute the If block. Otherwise, it will execute the Else block.

Syntax:

```
if(condition){  
    // statements  
}  
else{  
    // statements  
}
```

Example:

```
public class Main {  
  
    public static void main(String[] args) {  
  
        int time = 20;  
  
        if (time < 18) {  
  
            System.out.println("Good day.");  
  
        }  
  
        else {  
  
            System.out.println("Good evening.");  
  
        }  
  
    }  
  
}
```

Output: Good evening.

Note: In the example above, time (20) is greater than 18, so the condition is false. Because of this, we move on to the else condition and print "Good evening". If the time was less than 18, the program would print "Good day".

- **If Else-If statement**

If the condition is true, then it will execute the If block. Otherwise, it will execute the Else-If block. Again, if the condition is not met, then it will move to the else block.

Syntax:

```
if(condition1){  
    // statements  
  
}  
  
else if(condition2){  
    // statements  
}  
  
else{  
    //statements  
}
```

Example:

```
public class Main {  
  
    public static void main(String[] args) {  
  
        int time = 22;  
  
        if (time < 10) {  
  
            System.out.println("Good morning.");  
  
        }  
  
        else if (time < 20) {  
  
            System.out.println("Good day.");  
  
        }  
  
        else {
```

```

        System.out.println("Good evening.");
    }

}

}

```

Output: Good evening.

Note: In the example above, time (22) is greater than 10, so the first condition is false. The next condition, in the else if statement, is also false, so we move on to the else condition since condition1 and condition2 is both false - and print to the screen "Good evening".

However, if the time was 14, our program would print "Good day."

- **Switch Statement**

Switch statement allows program to select one action among multiple actions during the program execution.

Based on the argument in the switch statement suitable case value will be selected and executed.

If no matching case found, then the default will be executed.

It is optional to write a break statement at the end of each case statement.

If the default statement is used as the last statement in a switch block, it does not need a break.

Syntax:

```

switch(variable/value/expression)
{
    case value1: statement1
                break;
    case value2: statement2
                break;
    ...
    default: statement
}

```

Example:

```

public class Main {
    public static void main(String[] args) {
        int day = 2;
        switch (day) {
            case 1: System.out.println("Monday");
                    break;
            case 2: System.out.println("Tuesday");
                    break;
            case 3: System.out.println("Wednesday");

```

```
        break;
    case 4: System.out.println("Thursday");
        break;
    case 5: System.out.println("Friday");
        break;
    case 6: System.out.println("Saturday");
        break;
    case 7: System.out.println("Sunday");
        break;
    default: System.out.println("Invalid");
    }
}
```

Output: Tuesday

Note: If you write, `int day = 8;` then default statement will be executed and the output will be Invalid.