Flow Control Statement

- Conditional Statement
  - Branching
    - → Simple If
    - → If else
    - → else If ladder
    - → switch case
  - Looping
    - → while
    - → do while
    - → for
    - → for each
- Unconditional Statement
  - → break
  - → return
  - → continue

Branching Stmts (Decision Statement)

→ Simple If

→ If else

→ else If ladder

→ Switch case

→ Simple If :

statement can have only one option that we go for if statement

Syntax :

```
If (Condition)
{
 // Statement
}
```

Flow of program

1. Check the condition
2. If the condition gives (true) then control flow of the block executes statement and comes out of the block
3. If the condition false, control will go outside the block of the statement directly comes out of the block

Ex :

```
class M1
{
    public static void main(String[] args)
    {
        if true)
        {
            System.out.println(1);
            System.out.println(2);
            System.out.println(3);
            System.out.println(4);
        }
        System.out.println(5);
        System.out.println(6);
    }
}
```

O/P
1
2
3
4
5
6

The condition should always return boolean value (true or false)

If else : we make use of if else statement whenever we have two options that we need to execute one of it out any case.

Syntax :

```
If (Condition)
{
 //Stmt1,
}
else
{
 //Stmt2
}
```

OR

```
If (Condition)
 //Statement1,
else
 //Stmt2,
```

In if else statements, if we have to execute "statement" the condition needs to be true or else statement will get executed if the condition is false

```
class M1
{
    public static void main(String[] args)
    {
        System.out.println("Main Begin");
        int a = 10, b = 20;
        if (a<b)
        {
            System.out.println(1);
            System.out.println(2);
            System.out.println(3);
            System.out.println(4);
        }
        else
        {
            System.out.println(5);
            System.out.println(6);
        }
        System.out.println("Main End");
    }
}
```

Else If ladder :

Whenever we have more than 2 options and we need to execute one of them so we go for else if ladder.

Syntax :

```
If (Condition1) {
 //Stmt1
} else If (Condition2) {
 //Stmt2
}
else If (Condition3)
 //Stmt3
}
 ⋮
 ⋮
 ⋮
else {
 //Stmt4
}
```

Note : Inside else If ladder having other blocks is not mandatory

```
class M1
{
    public static void main(String[] args)
    {
        System.out.println("Main Begin");
        int a = 10, b = 20;
        if (a<b)
        {
            System.out.println(1);
            System.out.println(2);
            System.out.println(3);
            System.out.println(4);
        }
        else if(a==b)
        {
            System.out.println(5);
            System.out.println(6);
        }
        else if(a>b)
        {
            System.out.println(7);
            System.out.println(8);
        }
        else {
            System.out.println("Else");
        }
        System.out.println("Main End");
    }
}
```

* Inside else If ladder, only one block can get executed. Remaining multiple blocks are ignoring stmt.

```
class M1
{
    public static void main(String[] args)
    {
        System.out.println("Main Begin");
        int a = 10, b = 20;
        if (a<b)
        {
            System.out.println(1);
            System.out.println(2);
            System.out.println(3);
            System.out.println(4);
        }
        else if(a==b)
        {
            System.out.println(5);
            System.out.println(6);
        }
        else {
            System.out.println("Else");
        }
        System.out.println("Main End");
    }
}
```

3) The else block of executed only when all the if block values false.

```
class M1
{
    public static void main(String[] args)
    {
        System.out.println("Main Begin");
        int a = 10, b = 20;
        if (a<b)
        {
            System.out.println(1);
            System.out.println(2);
            System.out.println(3);
            System.out.println(4);
        }
        else {
            System.out.println(5);
            System.out.println(6);
        }
        System.out.println("Main End");
    }
}
```

Switch :

Syntax :

```
switch( value/expression)
{
 case value/Expression1:
  {
   //Statement,
  }
 case value/expression2:
  {
   //statement
   ⋮
   ⋮
   ⋮
  }
  default:
  {
   //stmt
  }
}
```

Note :

* In switch we can have the following data types. byte, short, int, char, string.
* The following data types cannot be passed. long, float, double, boolean
* Inside switch can have any number of cases
* The switch statement uses the predefined equality operator to compare the value of switch expression with each case
* If any of the cases satisfied to true, the block particular case statement is going to get executed or until we reach of the break. It will get executed irrespectively without more comparing the value

```
class M1
{
    public static void main(String[] args)
    {
        System.out.println("Main Begin");
        int a = 10;
        switch (a)
        {
        case 1:
        case 2:
        case 3:
        System.out.println(1);
        break;
        case 4:
        System.out.println(4);
        break;
        case 5:
        System.out.println(5);
        default:
        System.out.println("Default");
        break;
        }
        System.out.println("Main End");
    }
}
```

1) If execution satisfied then it goes out of the switch.

```
class M1
{
    public static void main(String[] args)
    {
        System.out.println("Main Begin");
        switch (2)
        {
        case 1:
        case 2:
        System.out.println(2);
        break;
        case 3:
        System.out.println(3);
        break;
        case 4:
        System.out.println(4);
        default:
        System.out.println("Default");
        break;
        }
        System.out.println("Main End");
    }
}
```

2) If execution satisfied then it goes out of the switch.

```
class M1
{
    public static void main(String[] args)
    {
        System.out.println("Main Begin");
        switch (2)
        {
        case 1:
        case 2:
        System.out.printlnln(2);
        case 3:
        System.out.printlnln(3);
        case 4:
        System.out.printlnln(4);
        default:
        System.out.printlnln("Default");
        }
        System.out.println("Main End");
    }
}
```

3) If execution satisfied then it goes out of the switch.

Else If ladder	Switch
else If ladder accept boolean	It doesn't accept boolean literals
Multiple conditions can be put in single else If ladder	We cannot have multiple conditions
The execution of block always depends on previous conditions	The execution of cases are independent