**Flow Control Statement:-**

They control flow of execution of the program. Flow control statements describes the order in which program statements will be executed. Flow control statements can be divided into 3 types:-

1. Decision making statements

if….else, switch

2. Looping statement

while, do...while, for

3. Jump statement

break, continue, return, try..catch..finally

**1. Decision Making statements-**

Decision-making statements decide which statement to execute. Decision-making statements evaluate the condition 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
* switch statement.

1. **If Statements :**

* Simple if statement
* if-else statement
* if-else-if ladder
* Nested if-statement
* **Simple if Statement :**

 It is used to execute the statement / statements when particular condition is true.

**Syntax :**

If (condition) {

//if the condition is true then if block will be executed

}

|  |
| --- |
| ***The condition must be of boolean type. If the expression is used then it must evaluate to boolean value. If it does not evaluate to boolean value then compile time error.*** |

**Example** :

|  |
| --- |
| ***public******class*** *Sample {*  ***public******static******void*** *main(String args[])*  *{*  ***int*** *x=10;*  ***int*** *y=20;*  ***if****(x+y)*  *{*  *System.****out****.println("Welcome");*  *}*  *}*  *}* |
| **Type mismatch: cannot convert from int to boolean** |

|  |
| --- |
| *public class SimpleIf {*  *public static void main(String[] args) {*  *int a=10 ;*  *int b= 20;*  *if(a<b) {*  *System.out.println("a is less than b");*  *}*  *}*  *}* |
| *Output:-*  *a is less than b* |

* **if-else statement :**

It 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){

//if the condition is true then if block will be executed

}

else{

//if the condition is false then else block will be executed

}

**Example** :

|  |
| --- |
| ***public******class*** *Student {*  ***public******static******void*** *main(String args[])*  *{*  ***int*** *marks=33;*  ***if****(marks>35)*  *{*  *System.****out****.println("You are pass");*  *}*  ***else***  *{*  *System.****out****.println("You are fail");*  *}*  *}*  *}* |
| Output:-  ***You are fail*** |

* **if-else-if ladder :**

**The if-else-if statement** contains the if-statement followed by multiple else-if statements.

**Syntax :**

if (condition 1){

statement1;//executes when condition1 is true

}

else if(condition 2) {

statement2;//executes when condition2 is true

}

else{

statement3;//executes when all conditions are false

}

**Example :**

|  |
| --- |
| **public** **class** Student {  **public** **static** **void** main(String args[])  {  **int** percentage = -10;  **if**(percentage>=66 && percentage<=100) {  System.***out***.println("Disctinction");  }  **else** **if**(percentage >=60 && percentage<66) {  System.***out***.println("First class");  }  **else** **if**(percentage>=55 && percentage<60) {  System.***out***.println("Higher second class");  }  **else** **if**(percentage>=40 && percentage<55)  {  System.***out***.println("Pass class");  }  **else** **if**(percentage>=0 && percentage<40)  {  System.***out***.println("Fail");  }  **else**  {  System.***out***.println("Invalid input");  }  }  } |
| *Output:-*  *Invalid input* |

* **Nested if-statement :**

If statement within other if statement is called as nested if statement.

**Syntax :**

if(condition1){

statement1;//executes when condition1 is true

if(condition 2)

{

statement2;//executes when condition2 is true

}

else

{

statement3;//executes when condition2 is false

}

}

**Example :**

**public** **class** NestedIf{

**public** **static** **void** main(String args[])

{

**int** a=10;

**int** b=20;

**if**(a==10){

**if**(b!=20){

System.***out***.println("Nested if executed");

}

**else**{

System.***out***.println("nested else executed");

}

}

}}

**Output :**

nested else executed.

* **Switch Statements:**

If multiple options are available, then instead of using if-else multiple times we should go for switch statement. The code becomes readable due to switch case statement.

**Syntax:**

switch(expression){

case 1:

Statement 1;

break;

Case 2:

Statement 2;

break;

default :

statement n;

}

**Example :**

**public** **class** SwitchStatement {

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

**int** day = 10;

String dayString ;

**switch**(day) {

**case** 1 :

dayString ="Monday";

**break**;

**case** 2:

dayString ="Tuesday";

**break**;

**case** 3:

dayString ="Wednesday";

**break**;

**case** 4:

dayString ="Thursday";

**break**;

**case** 5:

dayString ="Friday";

**break**;

**case** 6:

dayString ="Saturday";

**break**;

**case** 7:

dayString ="Sunday";

**break**;

**default** :

dayString = "Invalid day";

}

System.***out***.println(dayString);

}

}

**Points to remember:-**

1. If no case is matched then default case will be executed.

2. Case and defaults are optional.

3. Duplicate case labels are not allowed.

4. Within the switch statement if any case is matched from that case onwards all statements will be executed until end of the switch (or) break.

5. Within the switch we can take the default anywhere, but it is convention to take default as last case