

Agenda

Type casting in Java

If Statement

If Else Statement

Type Casting in Java

Assigning a value of one type to a variable of another type is known as Type Casting.

2 two types of casting in Java:

- Widening Casting(Implicit/ Automatic) - converting lower data type into higher.
- Narrowing Casting(Explicit/ Manual) - converting higher data type into lower.

Type Casting in Java

- Widening Casting(Implicit)

byte → short → int → long → float → double
—————→
widening

- Narrowing Casting(Explicitly done)

double → float → long → int → short → byte
—————→
Narrowing

```
1 package com.class1;
2
3 public class Casting {
4
5     public static void main(String[] args) {
6
7         double d = 10;
8
9         System.out.println(d);
10
11         double x = 2.7;
12         int y = (int) x;
13
14         System.out.println(y);
15     }
16 }
```

Console

<terminated> Modulus [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_144.jdk/Contents/Home/bin/java

10.0

2

Operators in Java

Relational & Equality Operators

OPERATOR	MEANING	EXAMPLE	RESULT
<	Less than	1<2	True
>	Greater than	1>2	False
<=	Less than or equal to	1<=2	True
>=	Greater than or equal to	1>=2	False
==	Equal to	1==2	False
!=	Not equal to	1!=2	True

Control Flow Statements

Java Flow Control

- 1) Java Conditional Statements
- 2) Java Loop Statements

Code is generally executed from top to bottom, in the order that they appear.

Control flow statements, however, break up the flow of execution by employing decision making, looping, and branching, enabling your program to conditionally execute particular blocks of code.

Java

Conditional Statements

Conditional Statements are used to insert verification points and error handling.

Two types of Conditional statements in Java

- 1) if statement**
- 2) switch statement**

if statement

```
If (I study hard) {
```



I got the job !

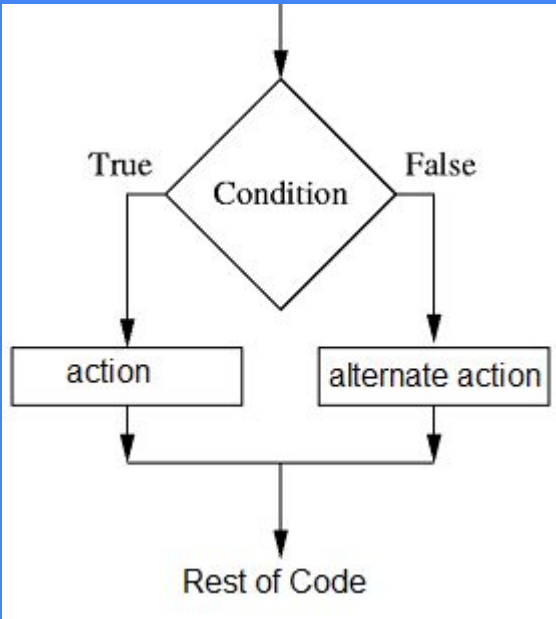
```
}else {
```



I did not get
the job !

```
}
```

if statement



This is the most basic of all the control flow statements, it tells your program to execute when a particular condition is true.

```
if(boolean_expression) {  
    // Executes when the Boolean expression is true  
}
```

Example:

```
if(a==2) {  
    System.out.println("The value of a is 2");  
}
```

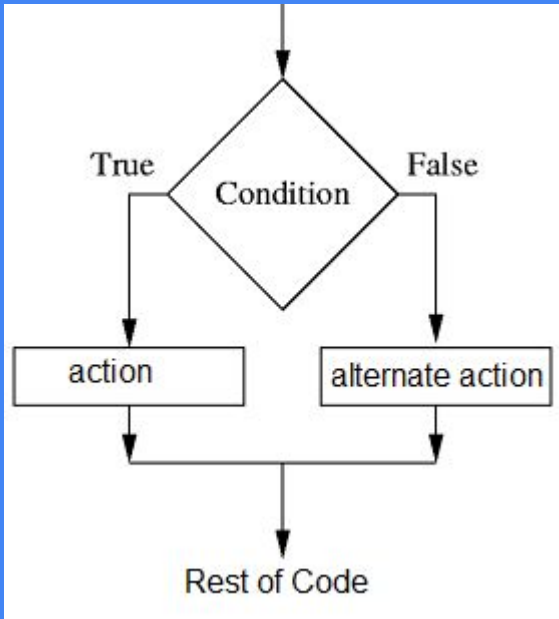

if ... else statement

This statement provides a secondary path of execution when an “if” condition is false it executes the command in else statement.

```
if(Boolean_expression) {  
    // Executes when the Boolean expression is true  
}else {  
    // Executes when the Boolean expression is false  
}
```

Example:

```
int a=2;  
if (a == 2) {  
    System.out.println(" the value of a is 2");  
} else {  
    System.out.println(" the value of a not 2");  
}
```



if...else...if Statement

This statement used to test various conditions

```
if(Boolean_expression 1) {  
    // Executes when the Boolean expression 1 is true  
}else if(Boolean_expression 2) {  
    // Executes when the Boolean expression 2 is true  
}else if(Boolean_expression 3) {  
    // Executes when the Boolean expression 3 is true  
}else {  
    // Executes when the none of the above condition  
    is true.  
}
```

if...else...if Statement

```
int day=3;

if (day==1) {
    System.out.println("Monday");
}else if(day==2){
    System.out.println("Tuesday");
}else if(day==3){
    System.out.println("Wednesday");
}else if(day==4){
    System.out.println("Thursday");
}else if(day==5){
    System.out.println("Friday");
}else if(day==6){
    System.out.println("Saturday");
}else if(day==7){
    System.out.println("Sunday");
}
```