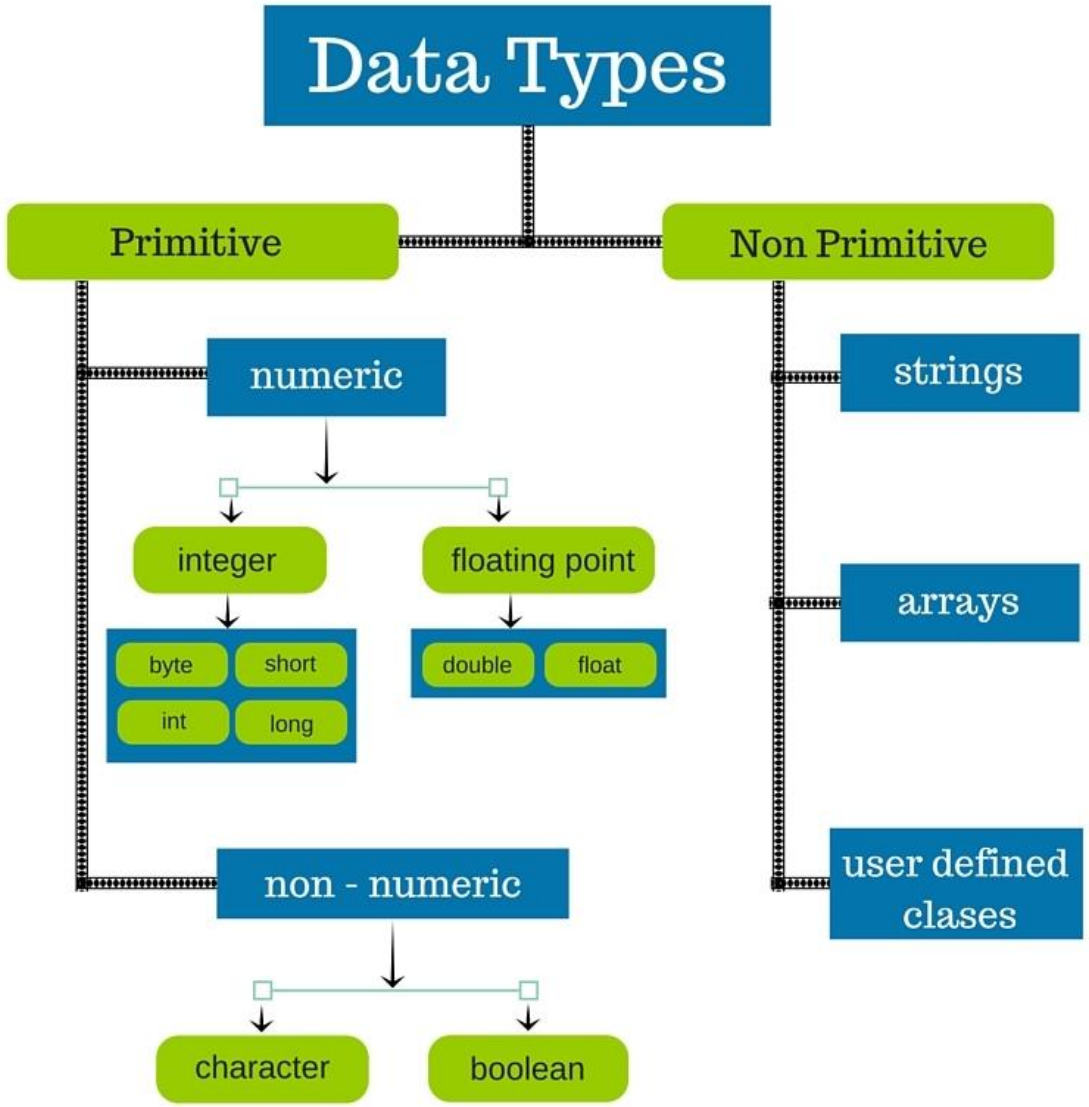
Java Programs

By
Dr. Radwa Rady& Dr. Ghada Fathy

Lecture Objectives

- Review
- Data types
- Data casting
- Operators
- User Input
- Examples.
- Conditional Statements:
  - ✓ IF
  - ✓ Switch
  - ✓ Ternary Operator
  - ✓ Nested Conditions
- Examples

Data Types


Data Types
Primitive
Non Primitive
numeric
integer
floating point
byte
short
int
long
double
float
non - numeric
character
boolean
strings
arrays
user defined clases

Type Casting

Type casting is when you assign a value of one primitive data type to another type. In Java, there are two types of casting:

- **Widening Casting** (**automatically**) - converting a smaller type to a larger type size

  byte -> short -> char -> int -> long -> float -> double

- **Narrowing Casting** (**manually**) - converting a larger type to a smaller size type

  double -> float -> long -> int -> char -> short -> byte

Widening Casting

```
public class Test {
    public static void main(String[] args) {

        int myInt = 9;
        double myDouble = myInt; // Automatic casting: int to double

        System.out.println(myInt);      // Outputs 9
        System.out.println(myDouble);   // Outputs 9.0

    }
}
```

Narrowing Casting

```
public class Test {
    public static void main(String[] args) {

        double myDouble = 9.78;
        int myInt = (int) myDouble; // Manual casting: double to int

        System.out.println(myDouble);   // Outputs 9.78
        System.out.println(myInt);      // Outputs 9

    }
}
```

User Input (Scanner)

```java
import java.util.Scanner;

public class Test {
    public static void main(String[] args) {

        Scanner myObj = new Scanner(System.in);

        System.out.println("Enter name, age and salary:");

        // String input
        String name = myObj.nextLine();

        // Numerical input
        int age = myObj.nextInt();
        double salary = myObj.nextDouble();

        // Output input by user
        System.out.println("Name: " + name);
        System.out.println("Age: " + age);
        System.out.println("Salary: " + salary);
      }
}
```

output

```
Enter name, age and salary:
Ali
21
10000
Name: Ali
Age: 21
Salary: 10000.0
```

User Input (Scanner)

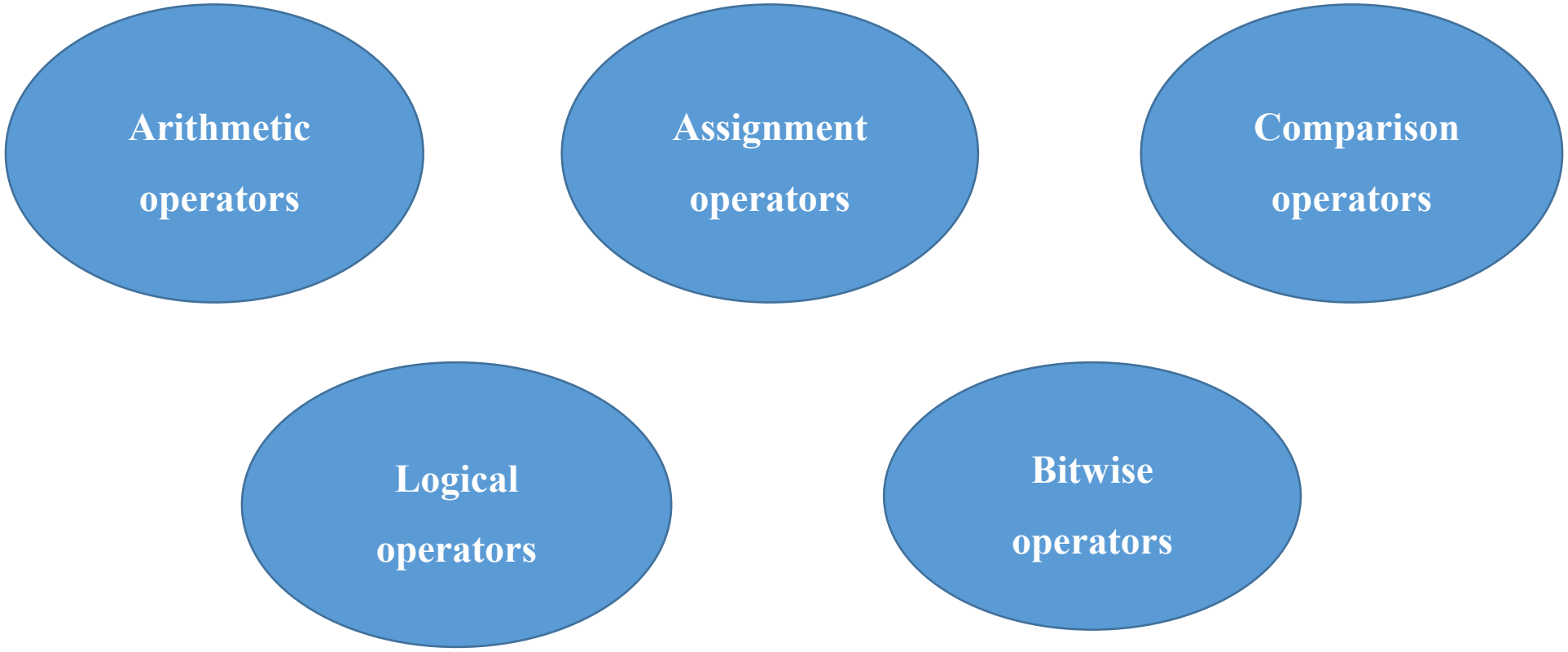
The Scanner class is used to get user input, and it is found in the java.util package. To use the Scanner class, create an object of the class and use any of the available methods found in the Scanner class documentation.

Input Types

Method	Description
nextBoolean()	Reads a boolean value from the user
nextByte()	Reads a byte value from the user
nextDouble()	Reads a double value from the user
nextFloat()	Reads a float value from the user
nextInt()	Reads a int value from the user
nextLine()	Reads a String value from the user
nextLong()	Reads a long value from the user
nextShort()	Reads a short value from the user

Operators

Operators are used to perform operations on variables and values. Java divides the operators into the following groups:



Arithmetic Operators

Arithmetic operators are used to perform common mathematical operations.

Operator	Name	Description	Example
+	Addition	Adds together two values	x + y
-	Subtraction	Subtracts one value from another	x - y
*	Multiplication	Multiplies two values	x * y
/	Division	Divides one value by another	x / y
%	Modulus	Returns the division remainder	x % y
++	Increment	Increases the value of a variable by 1	++x
--	Decrement	Decreases the value of a variable by 1	--x

Assignment Operators

Assignment operators are used to assign values to variables.

Operator	Example	Same As
=	x = 5	x = 5
+=	x += 3	x = x + 3
-=	x -= 3	x = x - 3
*=	x *= 3	x = x * 3
/=	x /= 3	x = x / 3
%=	x %= 3	x = x % 3
^=	x ^= 3	x = x ^ 3

Comparison Operators

Comparison operators are used to compare two values (or variables). The return value of a comparison is either true or false.

Operator	Name	Example
==	Equal to	x == y
!=	Not equal	x != y
>	Greater than	x > y
<	Less than	x < y
>=	Greater than or equal to	x >= y
<=	Less than or equal to	x <= y

Logical Operators

You can also test for true or false values with logical operators. Logical operators are used to determine the logic between variables or values.

Operator	Name	Description	Example
&&	Logical and	Returns true if both statements are true	x < 5 && x < 10
||	Logical or	Returns true if one of the statements is true	x < 5 || x < 4
!	Logical not	Reverse the result, returns false if the result is true	!(x < 5 && x < 10)

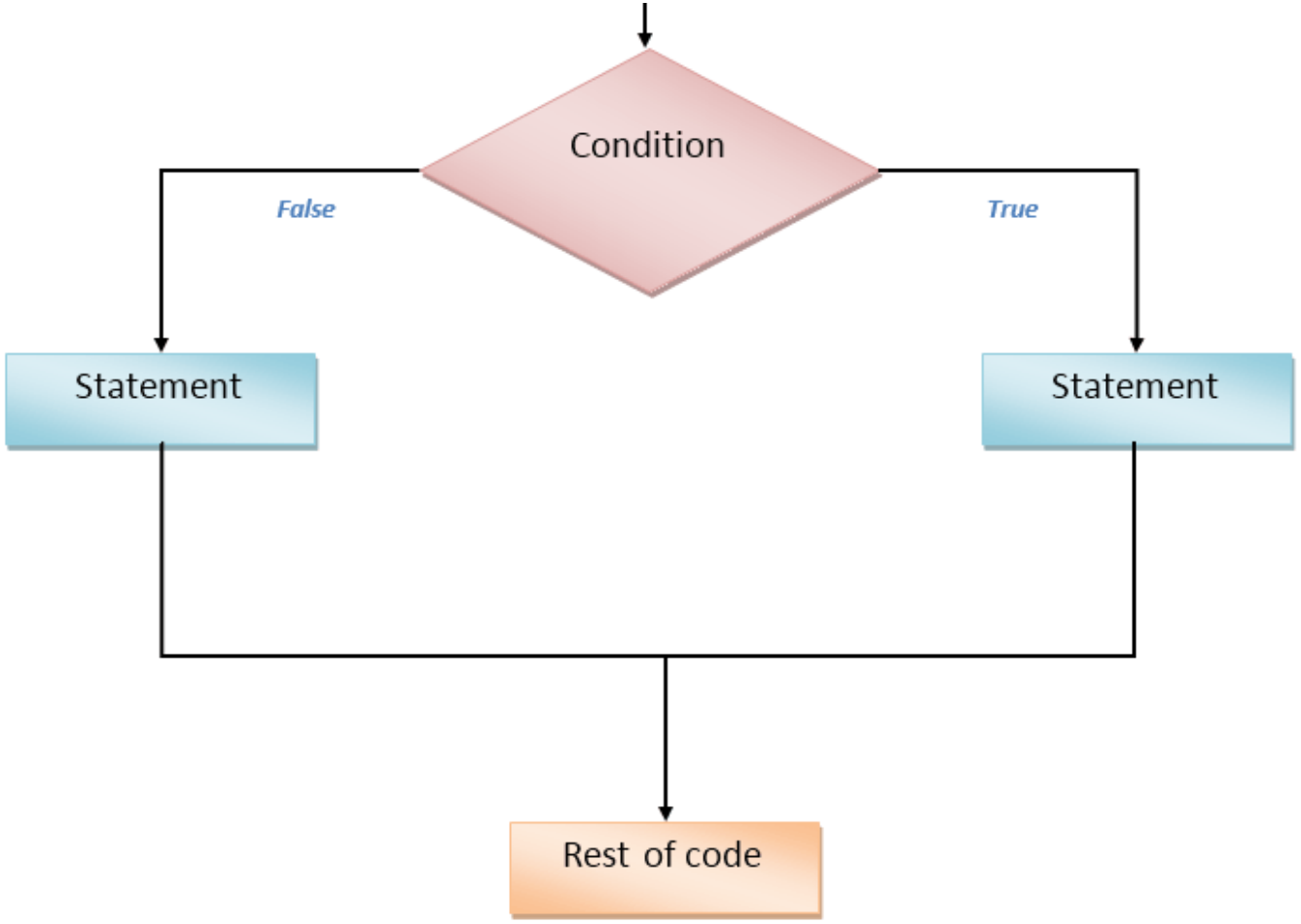
Example

Let's think of a "real life example" where we need to find out if a person is old enough to vote.

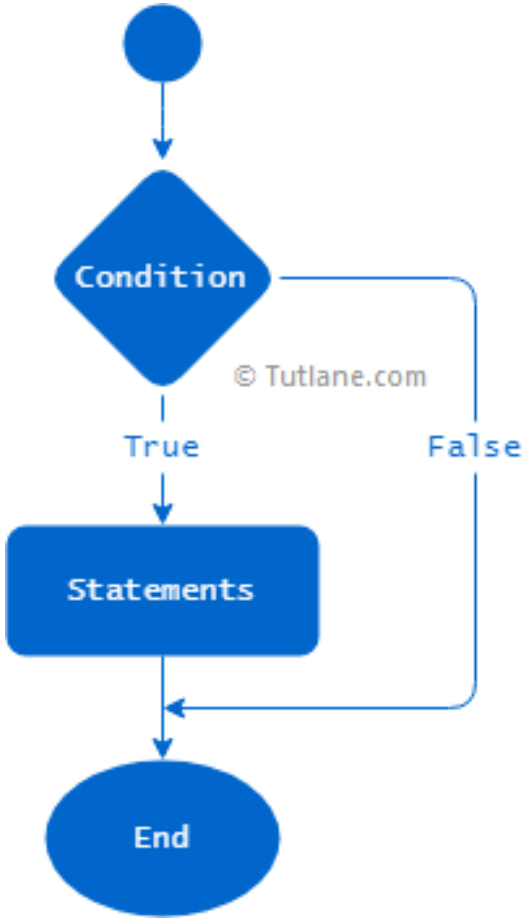
```
import java.util.Scanner;

public class Test {
    public static void main(String[] args) {

        int votingAge = 20;

        //int personAge = 25;

        System.out.print("Please enter your age: ");

        Scanner scReadAge = new Scanner(System.in);          // Create a Scanner object
        int personAge= scReadAge.nextInt();                  // Read user input


        System.out.println(personAge >= votingAge);

    }
}
```

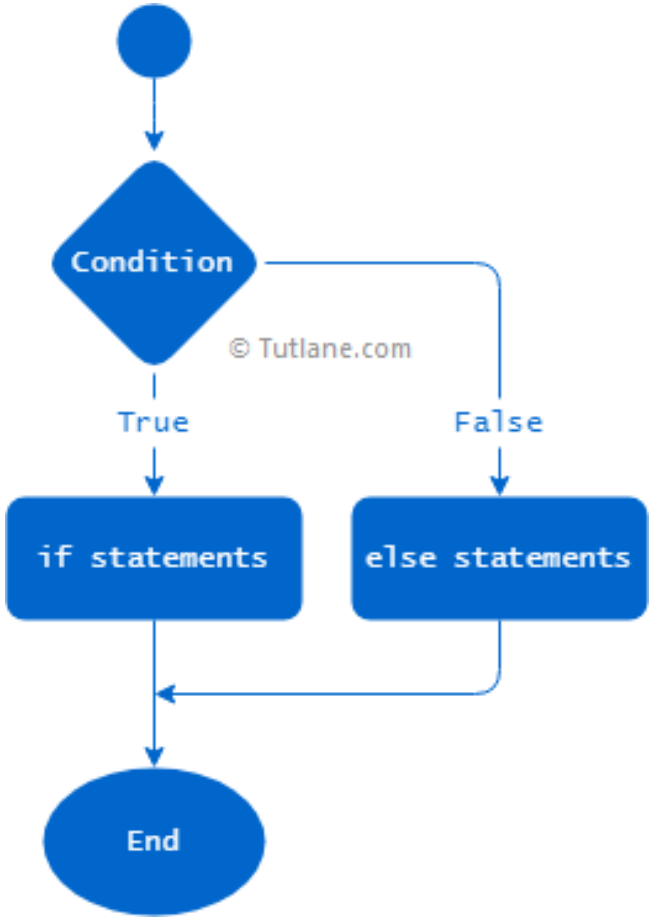
```
Please enter your age: 30
true
```

Conditional Statements



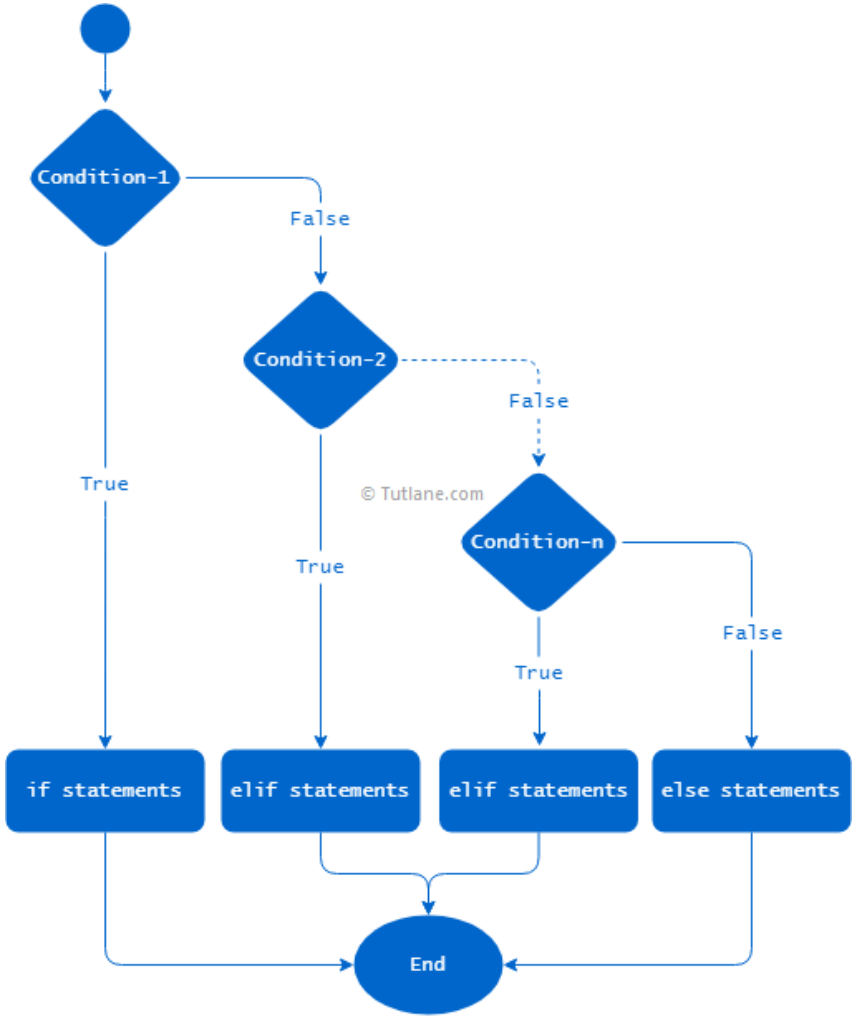
IF Conditional Statement





IF Conditional Statement

```
If (expression)
        {
        statements;
        }
else if (expression)
        {
        statements;
        }
else
        {
        statements;
        }
```



Example (Using If condition)

```java
import java.util.Scanner;

public class Test {
    public static void main(String[] args) {

        int myAge = 25;
        int votingAge = 18;

        if (myAge >= votingAge) {
          System.out.println("Old enough to vote!");
        } else {
          System.out.println("Not old enough to vote.");
        }

    }
}
```

output

```
Old enough to vote!
```

Example

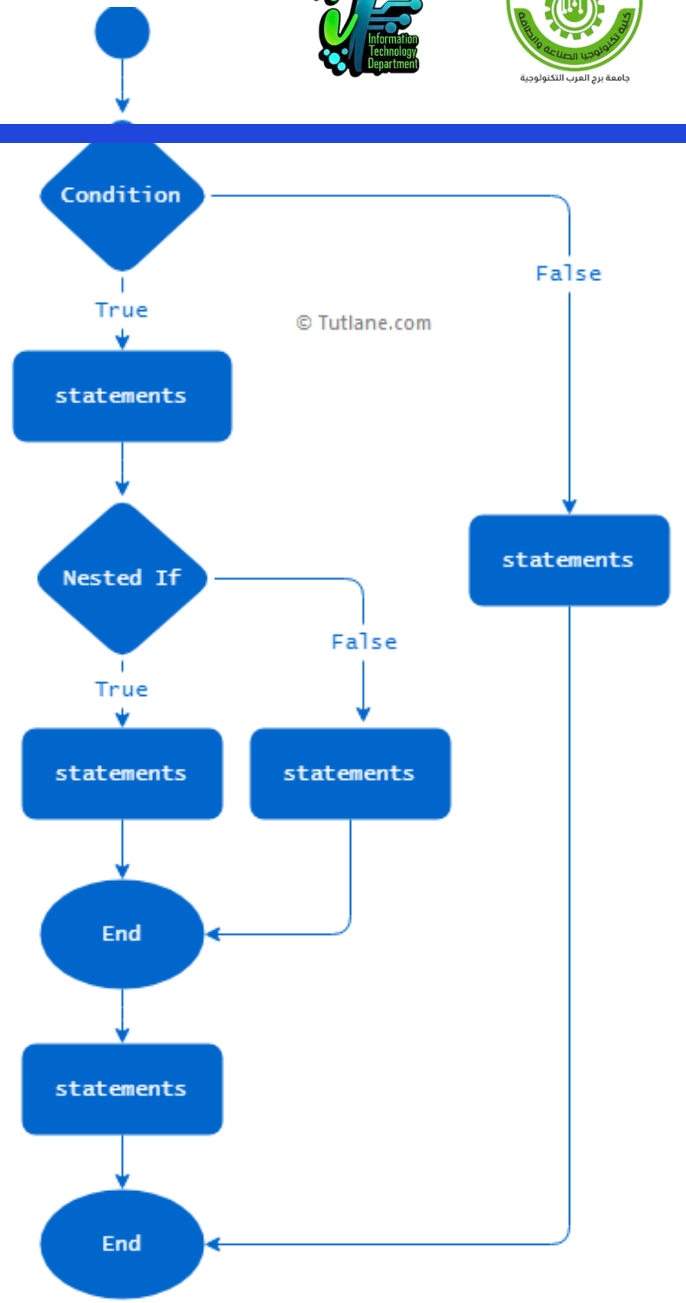
```
public class Test{

    public static void main (String[] args){

        int time = 22;   // Using 24 format.


        if (time < 10) {
        System.out.println("Good morning.");
        }
        else if (time < 18) {
        System.out.println("Good day.");
        }
        else {
        System.out.println("Good evening.");
        }
            }
}
```

output

```
Outputs "Good evening.
```

Nested IF Statements

```
if (expression1)
{           statement(s)
            if (expression2)
                        { statement(s) }
            else if (expression3)
                        { statement(s) }
            else
                        { statement(s) }
}
else:
            { statement(s) }
```



Example 1

```
public class Test{
    public static void main (String[] args){
        int x = 1010, y = 170, z = 169;
        if(x > y)
        {
            if(x > z)
            //prints x, if the above two conditions are true
            System.out.println("The largest number is: "+x);
            else
            //means x>y and x<z
            System.out.println("The largest number is: "+z);
        }
        else
        {
            if(y > z)
            //means z>x and y>z
            System.out.println("The largest number is: "+y);
            else
            //z>x and z>y
            System.out.println("The largest number is: "+z);
        }
    }
}
```

output

```
The largest number is: 1010
```

Example 2

```
public class Test
{
    public static void main (String[] args)
    {

    int a=40, b=78, c=19;

    if(a>b && a>c)
    System.out.println(a+" is the largest Number");

    else if (b>a && b>c)
    System.out.println(b+" is the largest Number");

    else
    //prints c if the above conditions are false
    System.out.println(c+" is the largest number");
    }
}
```

output

```
78 is the largest Number
```

Example 3

```java
import java.util.Scanner;

public class Test {
    public static void main (String[] args) {

        int num1, num2, num3;
        System.out.println("Enter three integers: ");
        Scanner sc = new Scanner(System.in);
        num1=sc.nextInt();
        num2=sc.nextInt();
        num3=sc.nextInt();

        if (num1 > num2 && num1 > num3)
        System.out.println("The largest number is: "+num1);
        else if (num2 > num1 && num2 > num3)
        System.out.println("The largest number is: "+num2);
        else if (num3 > num1 && num3 > num2)
        System.out.println("The largest number is: "+num3);
        else
        System.out.println("The numbers are same.");
    }
}
```

output

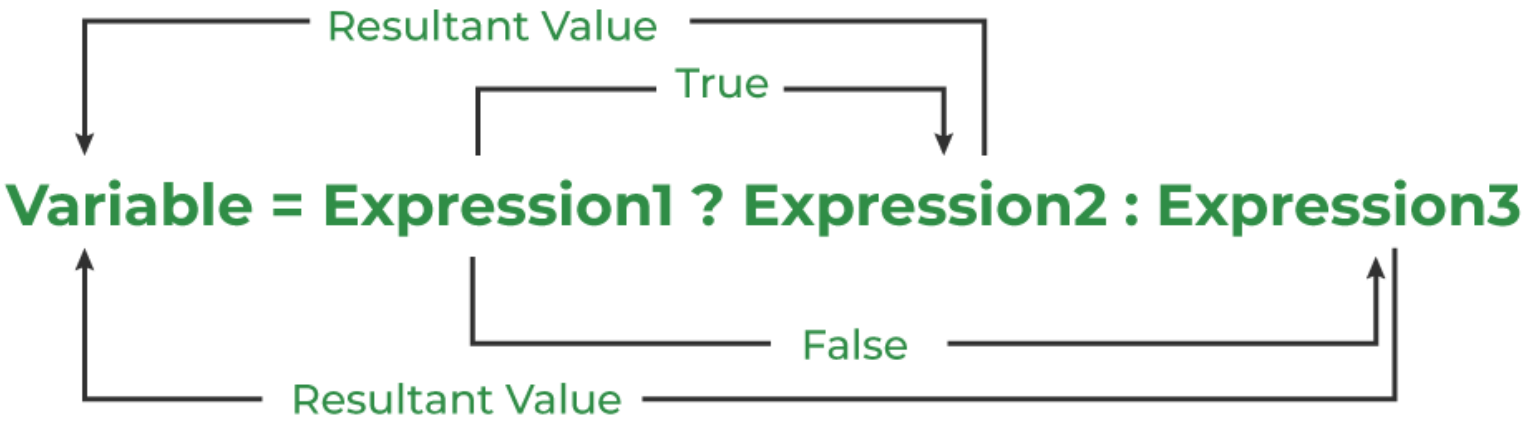
```
Enter three integers:
34
45
32
The largest number is: 45
```

Ternary Operator

There is also a short-hand if else, which is known as the ternary operator because it consists of three operands.

It can be used to replace multiple lines of code with a single line, and is most often used to replace simple if else statements.

```
variable = (condition) ? expressionTrue :  expressionFalse;
```



Example 1

```
public class Test
{
    public static void main (String[] args)
    {
        int time = 20;

        if (time < 18)
       {
          System.out.println("Good day.");
        }
       else
       {
          System.out.println("Good
evening.");
        }
    }
}
```

variable = (condition) ? expressionTrue : expressionFalse ;

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

Example 2

```java
import java.util.Scanner;

public class Test {
    public static void main (String[] args) {
        int a, b, c, largest, temp;
        Scanner sc = new Scanner(System.in);
        //reading input from the user
        System.out.println("Enter the first number:");
        a = sc.nextInt();
        System.out.println("Enter the second number:");
        b = sc.nextInt();
        System.out.println("Enter the third number:");
        c = sc.nextInt();

       //comparing a and b and storing the largest number in a temp variable
        temp=a>b?a:b;

       //comparing the temp variable with c and storing the result in the variable
        largest=c>temp?c:temp;

        System.out.println("The largest number is: "+largest);
    }
}
```

output

```
Enter the first number:
66
Enter the second number:
33
Enter the third number:
22
The largest number is: 66
```

Nested Ternary Operator

```
Temp = a > b ? a : b ;

Largest = c > temp ? C : temp;
```

```
Result = c > (a>b ? a:b)  ?  C : ((a>b) ? a:b);
```

Example 3

```java
import java.util.Scanner;

public class Test
{
    public static void main (String[] args)
    {
        int a, b, c, largest;
        //object of the Scanner class
        Scanner sc = new Scanner(System.in);
        //reading input from the user
        System.out.println("Enter the first number:");
        a = sc.nextInt();
        System.out.println("Enter the second number:");
        b = sc.nextInt();
        System.out.println("Enter the third number:");
        c = sc.nextInt();

        largest = c > (a > b ? a : b) ? c : ((a > b) ? a : b);
        System.out.println("The largest number is: "+largest);
    }
}
```

output

```
Enter the first number:
77
Enter the second number:
34
Enter the third number:
32
The largest number is: 77
```

Switch Statements

Instead of writing many if … else statements, you can use the switch statement. The switch statement selects one of many code blocks to be executed.

- The switch expression is evaluated once.
- The value of the expression is compared with the values of each case.
- If there is a match, the associated block of code is executed.
- The break and default keywords are optional.

```
switch(expression)
{
    case x:
      // code block
      break;
    case y:
      // code block
      break;
    default:
      // code block
 }
```

```java
public class Test {
    public static void main (String[] args) {
        int day = 4;
        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;
        }
    }
}
```

```
// Outputs "Thursday" (day 4)
```

Break Keyword

- When Java reaches a break keyword, it breaks out of the switch block.
- This will stop the execution of more code and case testing inside the block.
- When a match is found, and the job is done, it's time for a break. There is no need for more testing.

Default Keyword

The default keyword specifies some code to run if there is no case match:

```
public class Test {
    public static void main (String[] args) {

        int day = 4;                                    // Outputs "Looking forward to the Weekend"
        switch (day) {
          case 6:
            System.out.println("Today is Saturday");
            break;
          case 7:
            System.out.println("Today is Sunday");
            break;
          default:
            System.out.println("Looking forward to the Weekend");
        }

    }
}
```

Example 1

What will the output of this program?

```
public class Test {
    public static void main (String[] args) {

        int day = 6;
        switch (day) {
          case 6:
            System.out.println("Today is Saturday");
          case 7:
            System.out.println("Today is Sunday");
            break;
          default:
            System.out.println("Looking forward to the Weekend");
        }
    }
}
```

Output will be:

Today is Saturday

Today is Sunday

Example 2

What will the output of this program?

```
public class Test {
    public static void main (String[] args) {

        int day = 6;
        switch (day) {
          case 6:
            System.out.println("Today is Saturday");
          case 7:
            System.out.println("Today is Sunday");
          default:
            System.out.println("Looking forward to the Weekend");
        }
    }
}
```

Output will be:

Today is Saturday

Today is Sunday

Looking forward to the Weekend

Thanks

References: https://www.w3schools.com