SNS COLLEGE OF TECHNOLOGY

(An Autonomous Institution)
Coimbatore - 641035.

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A++" Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

Department of Computer Applications

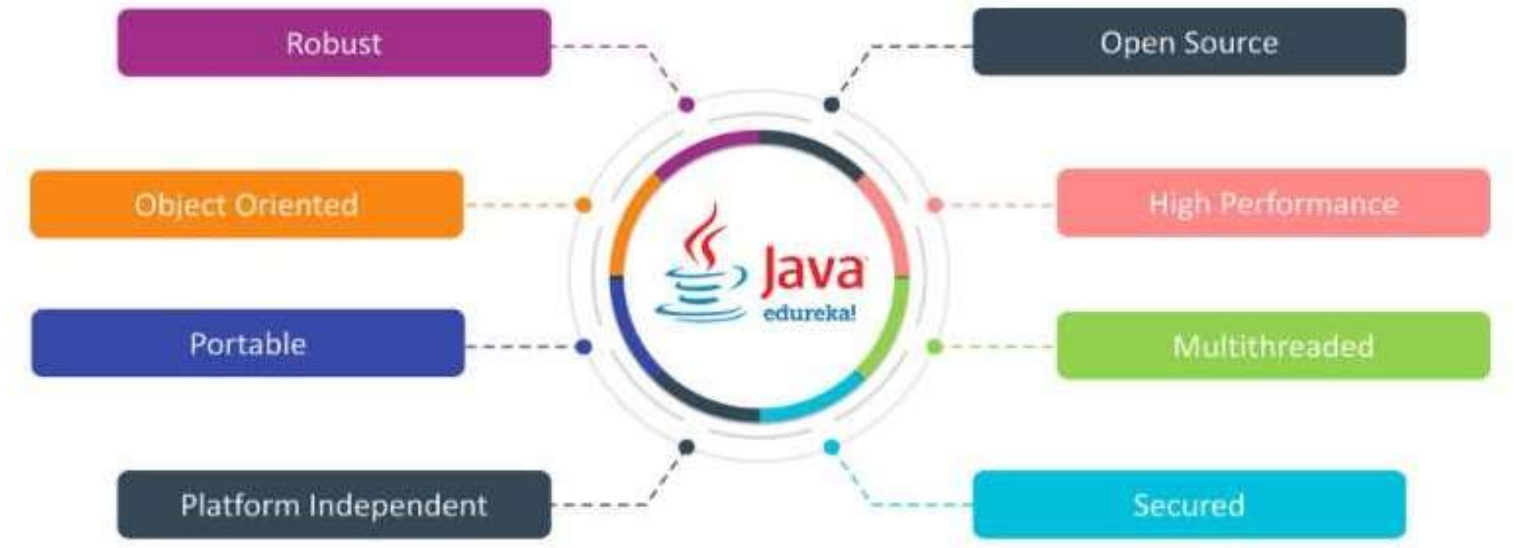
Course Code: 23CAT606

Course Name: Java Programming

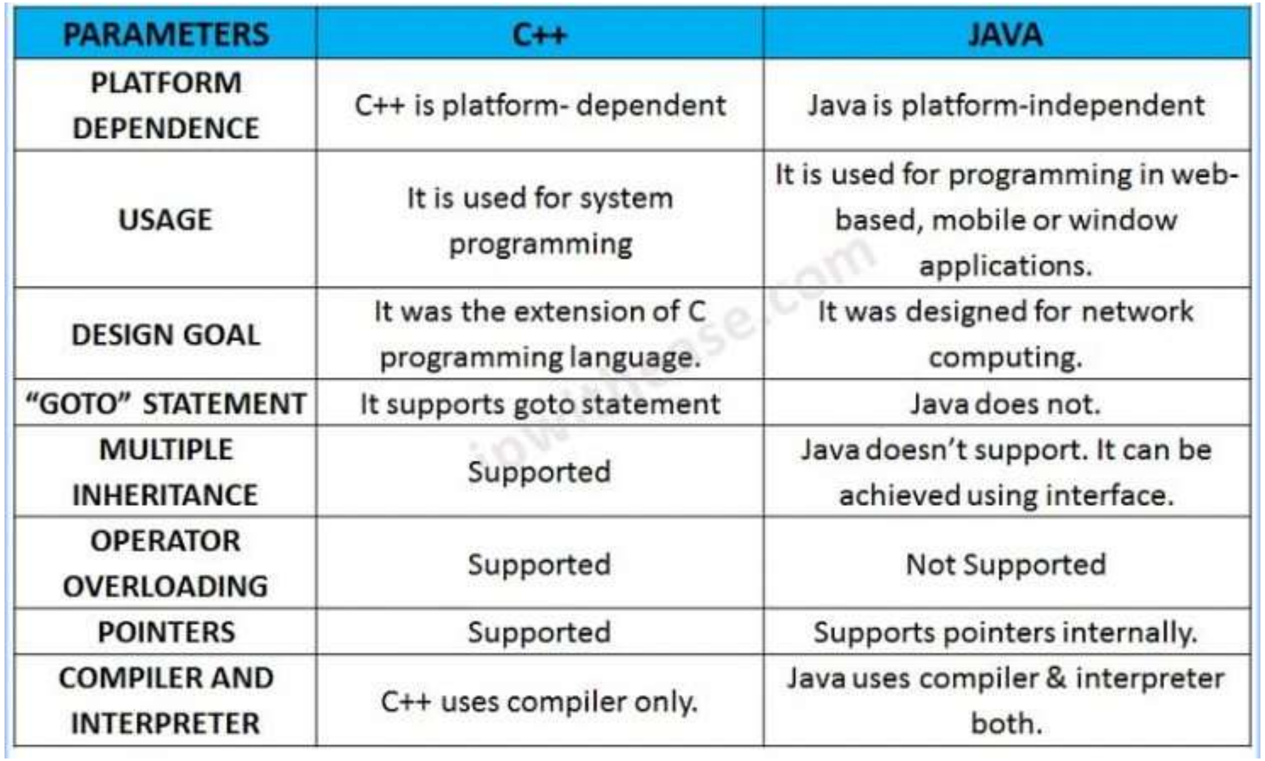
Unit I: Java Fundamentals

Topic 3 Java Fundamentals

Java Fundamentals: Features



Difference between C++ and Java



PARAMETERS	C++	JAVA
PLATFORM DEPENDENCE	C++ is platform- dependent	Java is platform-independent
USAGE	It is used for system programming	It is used for programming in web-based, mobile or window applications.
DESIGN GOAL	It was the extension of C programming language.	It was designed for network computing.
"GOTO" STATEMENT	It supports goto statement	Java does not.
MULTIPLE INHERITANCE	Supported	Java doesn't support. It can be achieved using interface.
OPERATOR OVERLOADING	Supported	Not Supported
POINTERS	Supported	Supports pointers internally.
COMPILER AND INTERPRETER	C++ uses compiler only.	Java uses compiler & interpreter both.

Simple Java Program

```
public class FirstProgram {
    public static void main(String[] args){
        System.out.println("Hello
        World");
    }
}
```

```
//Output:

Hello
World
```

Member Variables: A member variable plays a major role in a class as it is used to store a data value. When we define a class, we can declare a member variable. These variables are members of a class.

1. Local variable
2. Instance variable
3. Class/Static variable

Member Variables

Local Variable

```
public class Car {
    public void display(int m){  // Method
        int model=m;  // Created a local variable model
        System.out.println("Model of the car is"
        +model);
    }
}
```

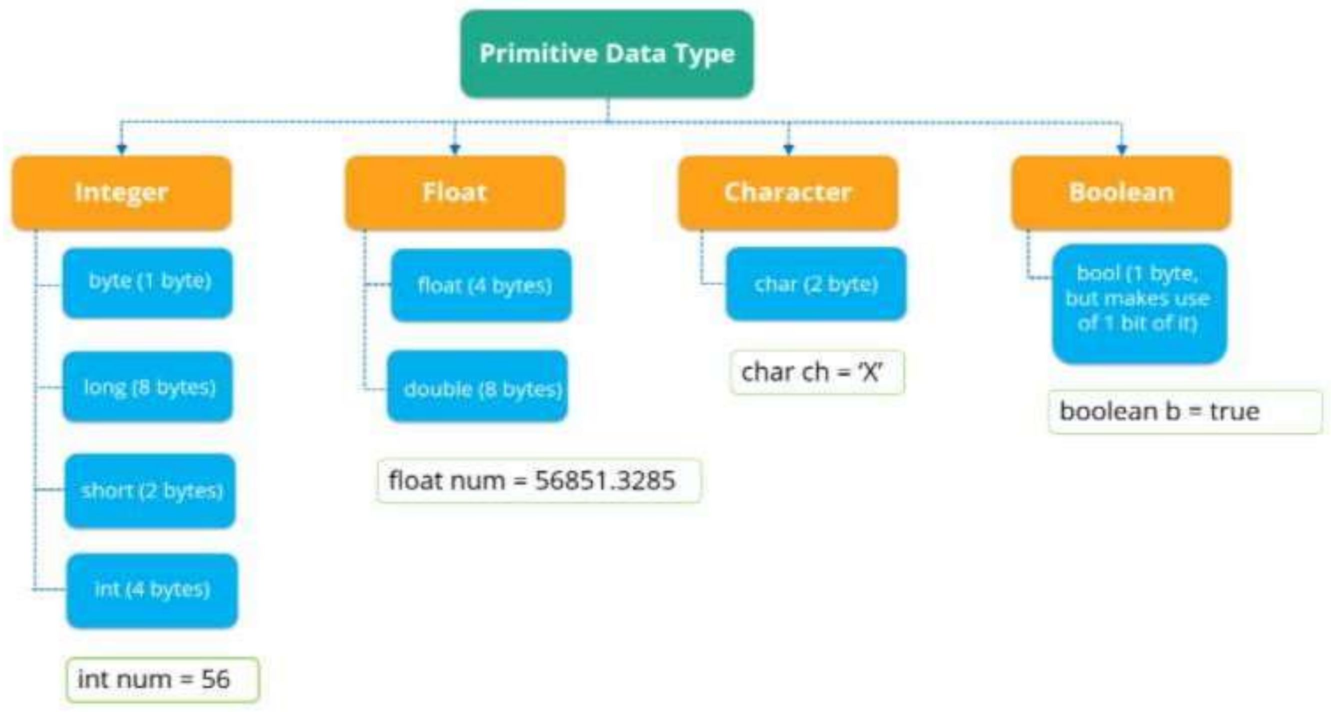
Instance Variable

```
public class Car {
    public String color;    // Created an instance variable color
    Car(String c){
        color=c;
    }
    public void display() {  // Method
        System.out.println("color of the car is"+color);
    }
public static void main(String args[]){
        Car obj=new Car("black");
            obj.display();
    }
}
```

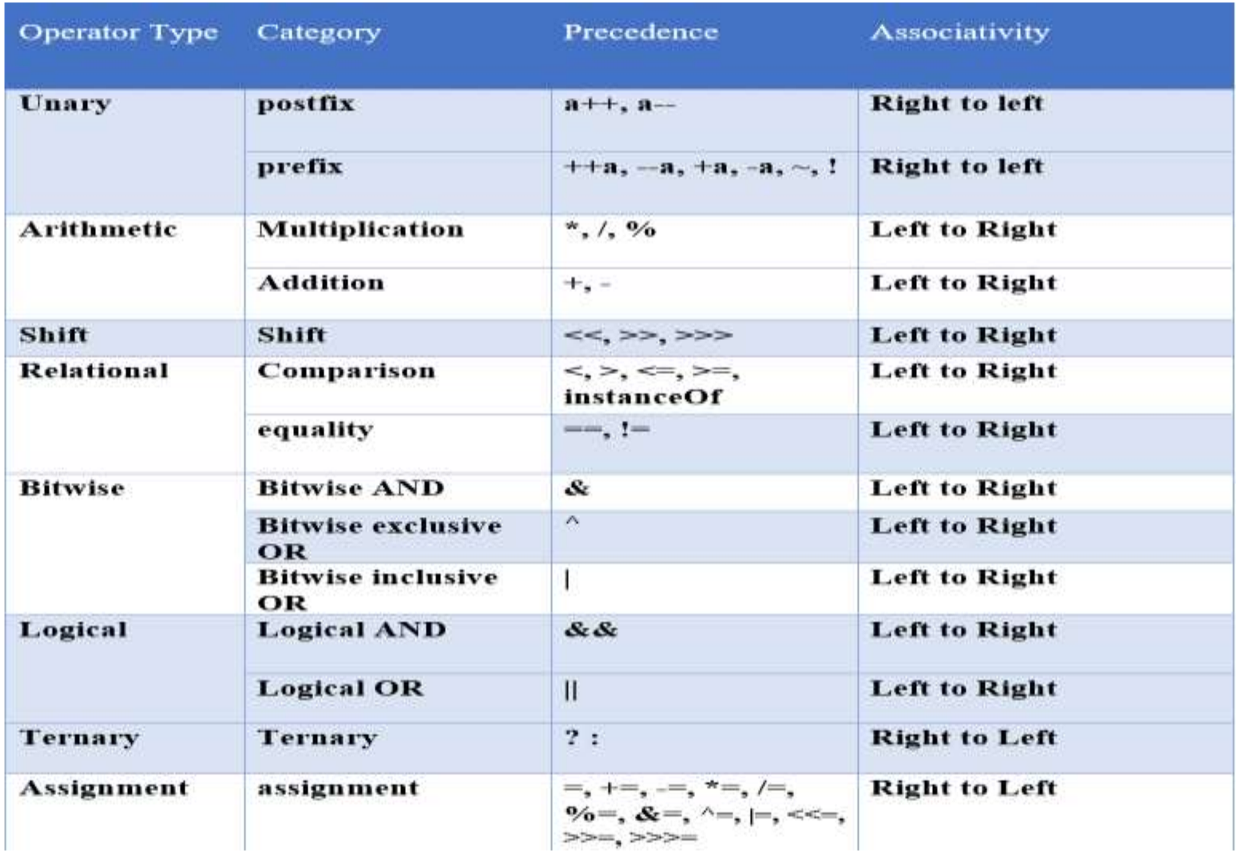
Class Variable

```
public class Car {
    public static int            // Created a class variable tyres
    tyres;
    public static void main(String args[]){
        tyres=4;
        System.out.println("Number of tyres
        are"+tyres);
}   }
```

Datatypes

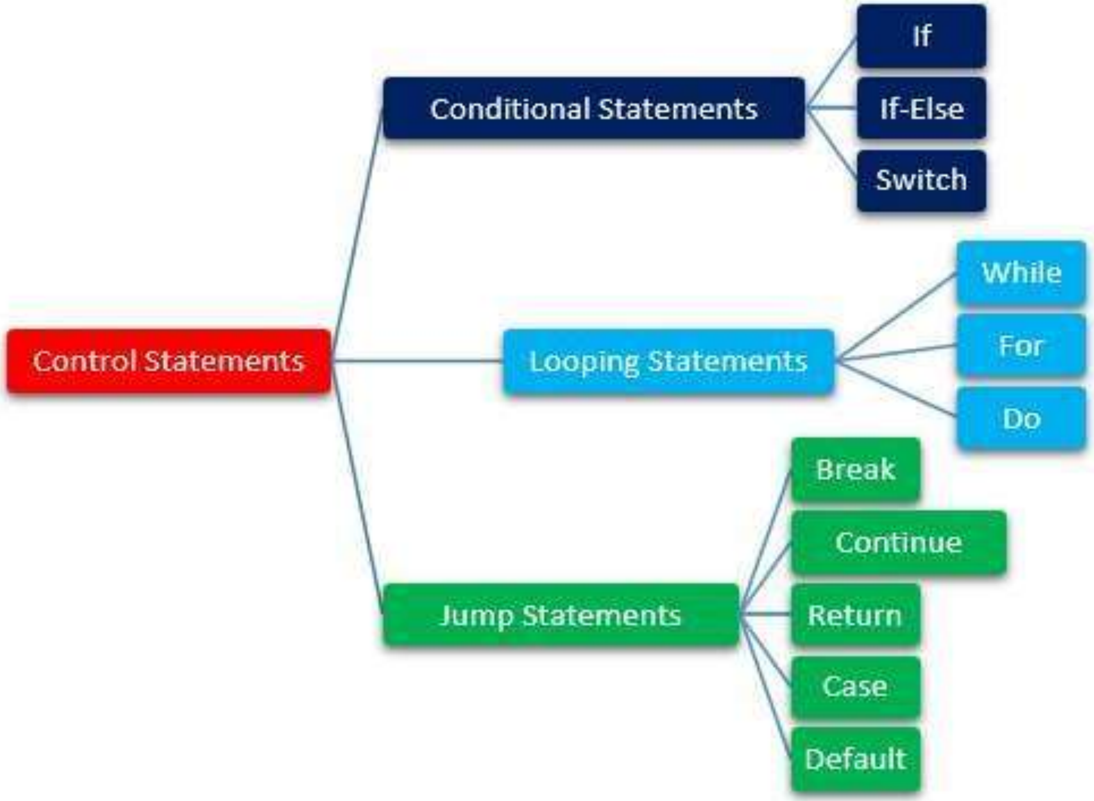

Primitive Data Type
Integer
byte (1 byte)
long (8 bytes)
short (2 bytes)
int (4 bytes)
int num = 56
Float
float (4 bytes)
double (8 bytes)
float num = 56851.3285
Character
char (2 byte)
char ch = 'X'
Boolean
bool (1 byte, but makes use of 1 bit of it)
boolean b = true

Data Operators



Operator Type	Category	Precedence	Associativity
Unary	postfix	a++, a--	Right to left
	prefix	++a, --a, +a, -a, ~, !	Right to left
Arithmetic	Multiplication	*, /, %	Left to Right
	Addition	+, -	Left to Right
Shift	Shift	<<, >>, >>>	Left to Right
Relational	Comparison	<, >, <=, >=, instanceOf	Left to Right
	equality	==, !=	Left to Right
Bitwise	Bitwise AND	&	Left to Right
	Bitwise exclusive OR	^	Left to Right
	Bitwise inclusive OR	|	Left to Right
Logical	Logical AND	&&	Left to Right
	Logical OR	||	Left to Right
Ternary	Ternary	? :	Right to Left
Assignment	assignment	=, +=, -=, *=, /=, %=, &=, ^=, |=, <<=, >>=, >>>=	Right to Left

Control statements



Control statements

1. **Conditional statements** are used to decide if a specific set of statements to be executed or skipped

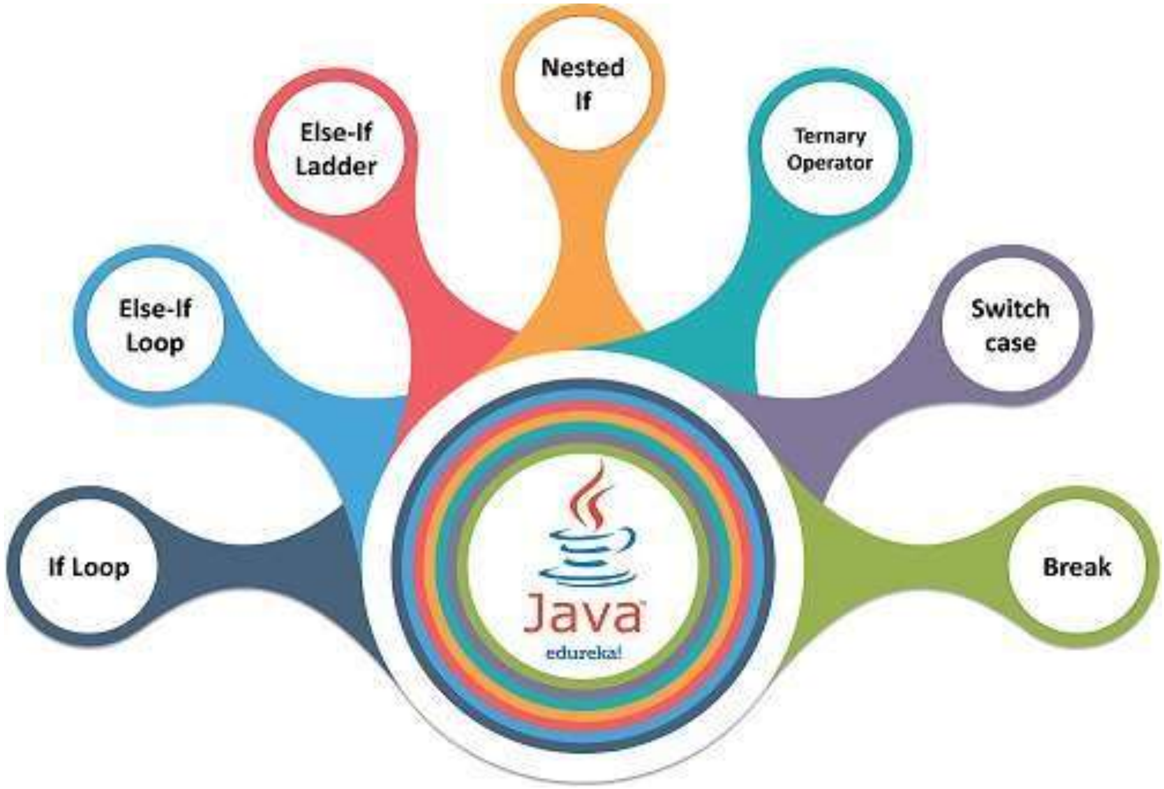
   based on the condition written in the program.
2. **Looping statements** are used to repeat specific set of statements until the condition specified

   remains true.

3. **Jumping statements** are used to abruptly exit from a particular statement, these are generally used in

   conjunction with conditional constructs.

Conditional statements

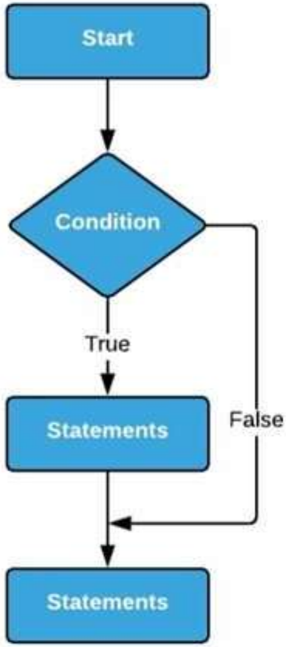


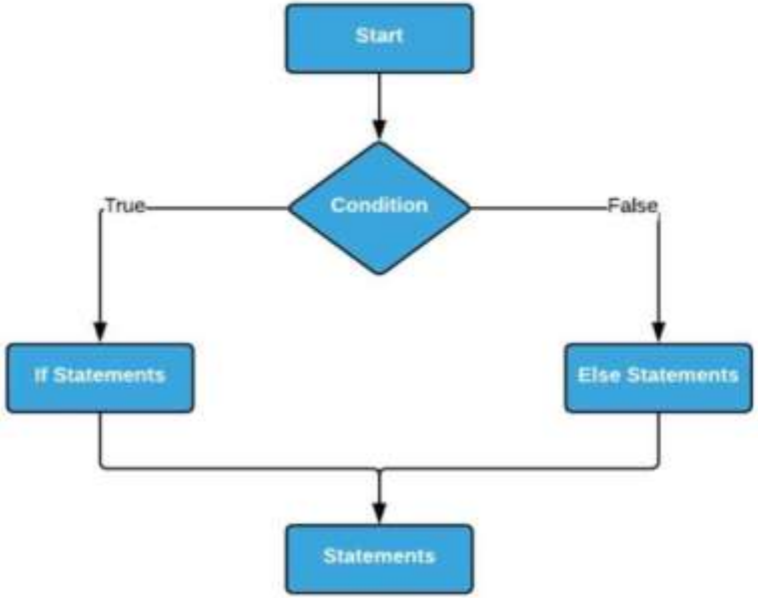
Conditional Statements

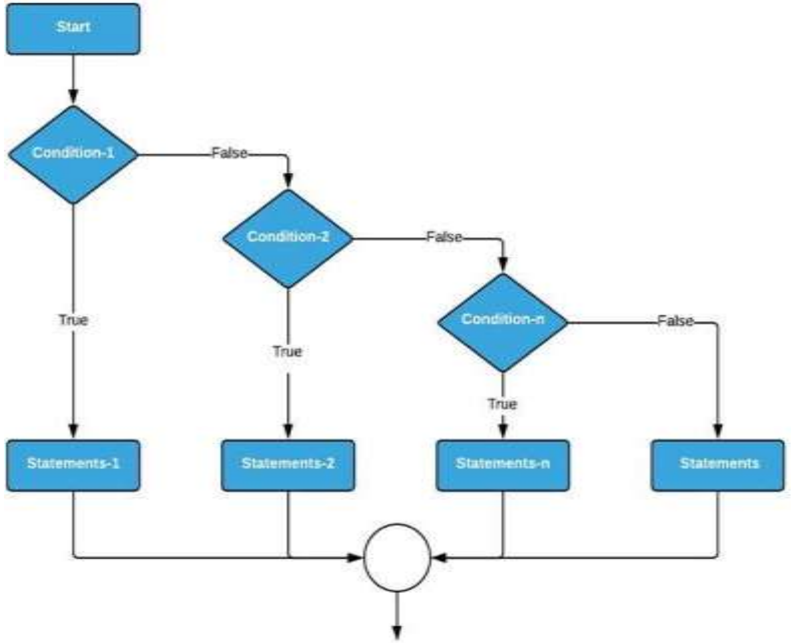
If Condition

Else-If Condition

Else-If Ladder

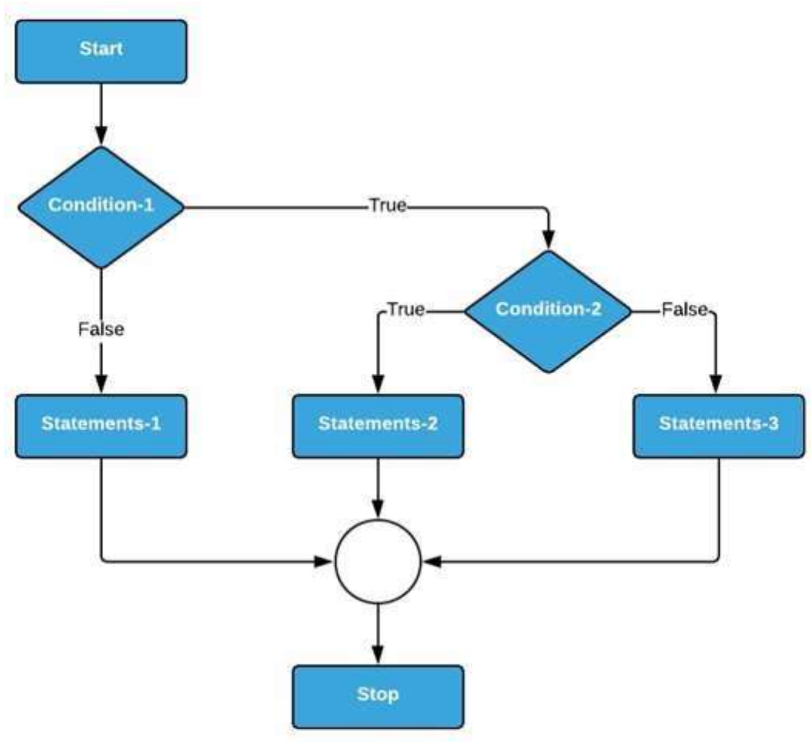




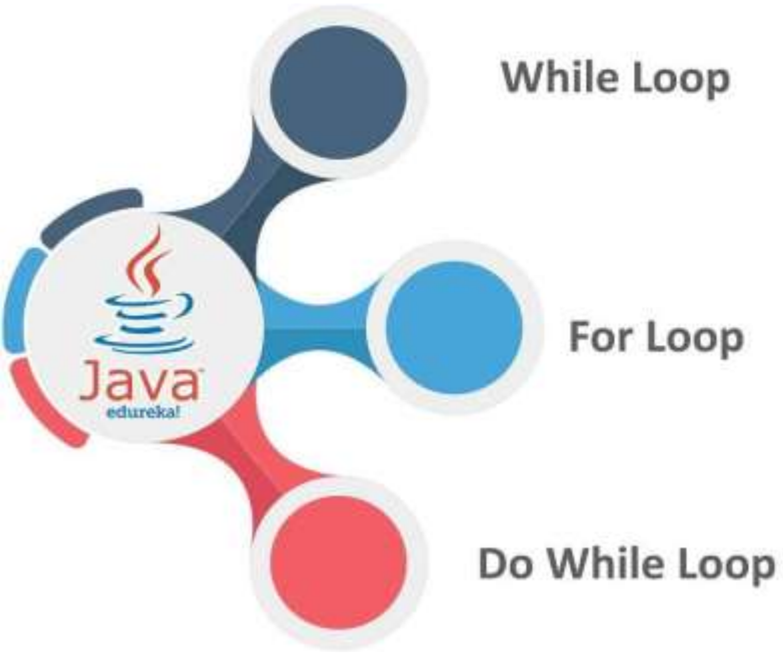


Conditional Statement

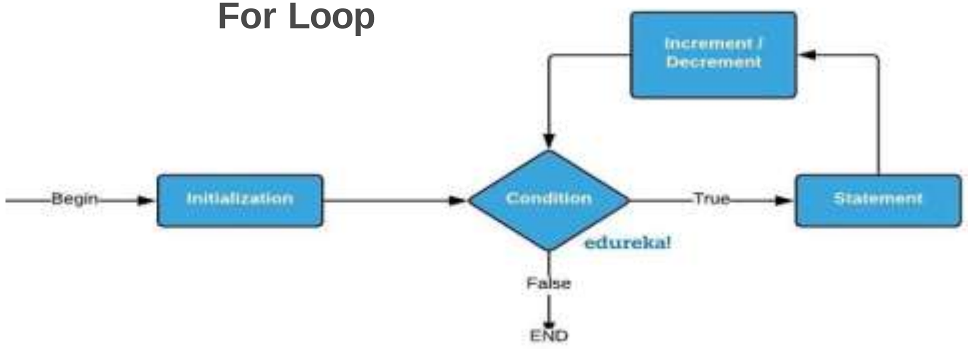
Nested-If



Looping Statements

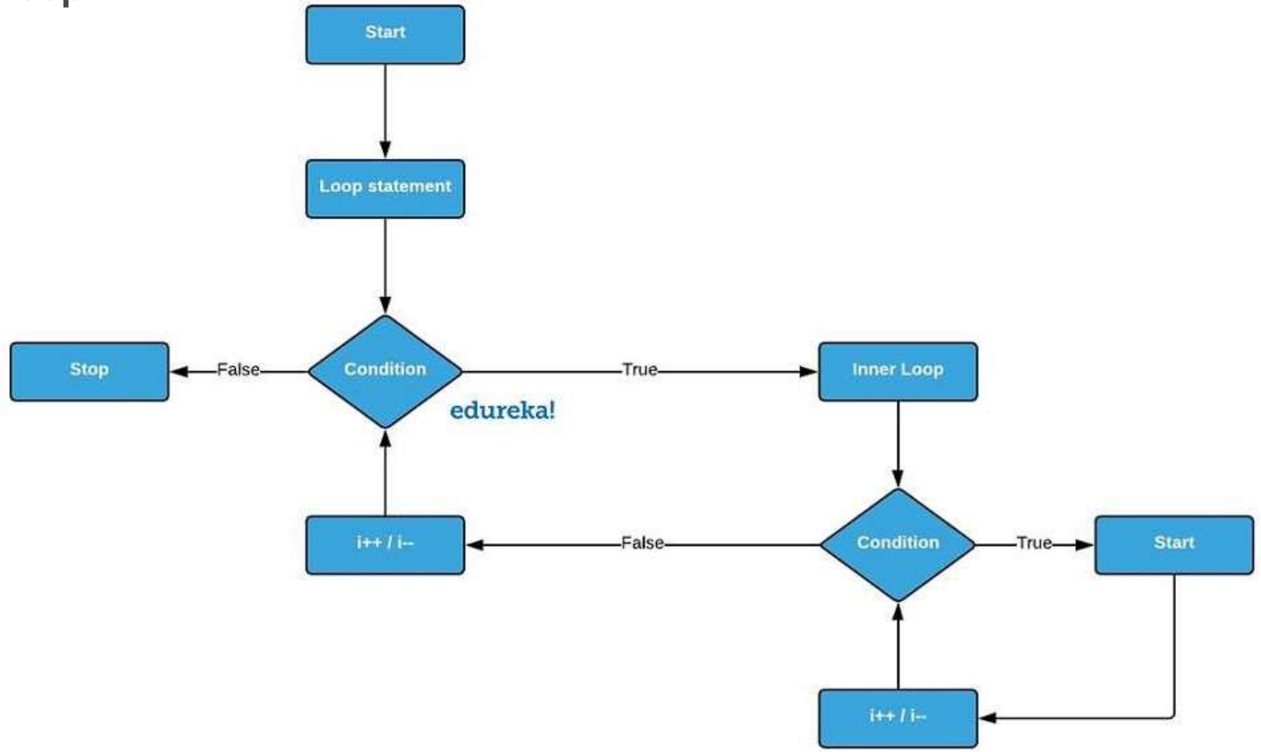


For Loop



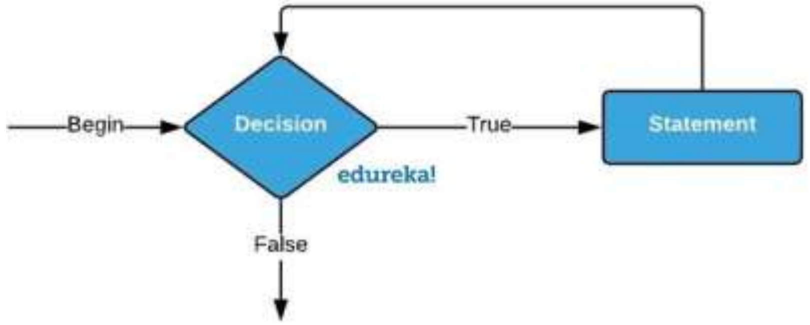
Looping Statements

Nested For Loop

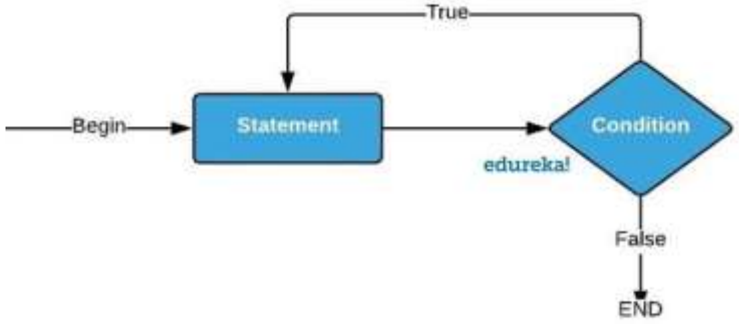


Looping Statements

While Loop



Do While Loop



Conditional Statement: Example

```
// sample if condition
if( i % 2 == 0 )
    {
        i = i * i;
    }


//sample if else condition
if(i%2==0)
    {
        i = i*i; // square of the number
    }
 else
    {
        i = i*i*i; // cube of the number
    }
```

```
public void printNumber(int num)
{
   switch(num)
   {
      case 0:     System.out.println(" Zero ");     break;
      case 1:     System.out.println(" One ");      break;
      case 2:     System.out.println(" Two");       break;
      case 3:     System.out.println(" Three");     break;
      case 4:     System.out.println(" Four");      break;
      case 5:     System.out.println(" Five");      break;
      case 6:     System.out.println(" Six");       break;
      case 7:     System.out.println(" Seven");     break;
      case 8:     System.out.println(" Eight ");    break;
      case 9:     System.out.println(" Nine");      break;
      default:    System.out.println("Invalid Number");
   }
}
```

Conditional Statement: Example

if-else-if ladder Statement

The if-else-if ladder statement executes one condition
from
multiple statements.

Syntax:

```
if(condition1){
//code to be executed if condition1 is true
}else if(condition2){
//code to be executed if condition2 is true
}
else if(condition3){
//code to be executed if condition3 is true
}
...
else{
//code to be executed if all the conditions are
false
}
```

```
public class IfElseIfExample {
public static void main(String[] args)
{
    int marks=65;
    if(marks<50){
        System.out.println("fail");
    }
    else if(marks>=50 && marks<60){
        System.out.println("D grade");
    }
    else if(marks>=60 && marks<70){
        System.out.println("C grade");
    }
    else if(marks>=70 && marks<80){
        System.out.println("B grade");
    }
    else if(marks>=80 && marks<90){
        System.out.println("A grade");
    }else if(marks>=90 &&
        marks<100){
        System.out.println("A+ grade");
    }else{
}       System.out.println("Invalid!");
}   }
```

Conditional Statement: Example

Java Nested if statement

```
if(condition){
     //code to be executed
          if(condition){
             //code to be executed
    }
}
```

```
public class JavaNestedIfExample {
public static void main(String[] args) {
//Creating two variables for age and weight
int age=20;
int weight=80;
//applying condition on age and weight
    if(age>=18){ if(weight>50){
    System.out.println("You are eligible to donate blood");
}
}
}}
```

Using Ternary Operator: Example

We can also use ternary operator (? :) to perform the task of if...else statement.

```
public class IfElseTernaryExample {
public static void main(String[] args) {
    int number=13;
    //Using ternary operator
    String output=(number%2==0)?"even number":"odd number";
    System.out.println(output);
}
}
```

```
Output: odd number
```

Looping Statement: Example

```
//sample while condition
while ( i < 10 )
{
i = i + 1;
System.out.println( "Value of i:" +
i);
}
```

```
//sample for loop
for (int i=0;i< 10;i++)
{
System.out.println("Value of i:"+
i);
}
```

```
//sample for do while
int i = 1;
// this loop will execute the statements inside the loop even
// if the condition does not evaluate to true for the first
time.
do
{
   System.out.println("Value of i:" + i) ;
   i++;
 } while( i < 1) ;
```

```
// sample for continue statement
for(int i=2;i<10;i++)
{
   System.out.println("I:" + i );
   if( i % 3 == 1) // if the value of i
   is
   {
      continue;
     // skips the steps for this iteration and moves to
     next
     // iteration in the loop
   }
   System.out.println("after statement");
}
```

```
// sample for break
statement for(int
i=2;i<10;i++)
{
   System.out.println("I:" + i );
   if( i % 3 == 1) // if the value of i is
   {
     break; // exits from the for loop
   }
   System.out.println("after
   statement");
}
```

Summary
1
2
3