Loops in java
==========

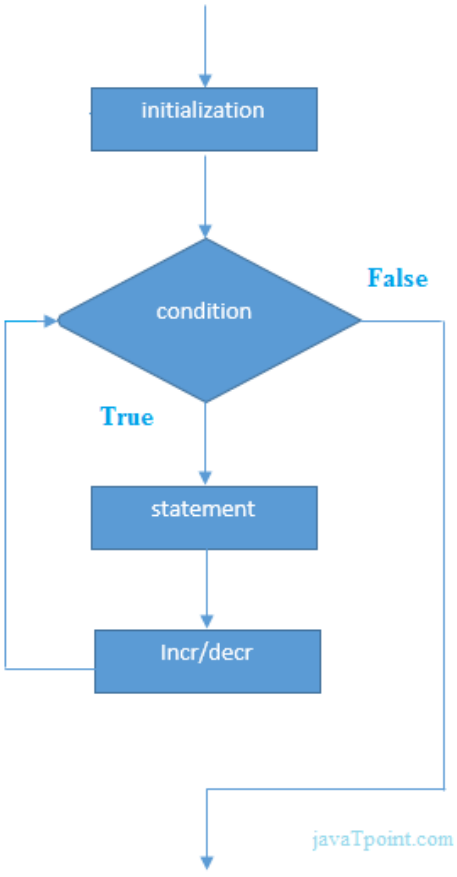
What is loops > when a programmer needs to perform repetitive tasks then we use loops concept. There are following types of loops

Types of loops

1. **For loop**
2. **While loop**
3. **Do while**

1. **For loop**

    a. **Single for loop**



for loop provides a compact way of writing the loop structure, a for statement consumes the initialization, condition and increment/decrement in one line.

1. **Initialization**: It is the initial condition which is executed once when the loop starts. Here, we can initialise the variable, or we can use an already initialised variable. It is an optional condition.

2. **Condition**: It is the second condition which is executed each time to test the condition of the loop. It continues execution until the condition is false. It must return a boolean value either true or false. It is an optional condition.
3. **Increment/Decrement**: It increments or decrements the variable value. It is an optional condition.
4. **Statement**: The statement of the loop is executed each time until the second condition is false.

Syntax >>

```
for ( initialization; condition;  increment/decrement)
{
        statement(s)
}
```

E.x

```
public class aclass
        {
        public static void main(String[] args)
                {
                //for(initialisation; condition; increment/decrement)
                for(int i = 7;    i > 5;    i--)
                {
                        System.out.println("print statement");
                }
        }
}
```

—----------

b. Nested for Loop

If we have a for loop inside another loop then it is known as nested for loop. The inner loop executes completely whenever the outer loop executes.

Syntax >>

```
for ( initialization; condition;  increment/decrement)
{
        statement(s)
                for ( initialization; condition;  increment/decrement)
                {
                statement(s)
                }
}
```

E.x

```
public class aclass
{
	public static void main(String[] args)
	{
		//for(initialisation; condition; increment/decrement)
		for(int i = 7;	i > 5;	i--)
		{
			System.out.println("print statement");

			for(int i = 7;	i > 5;	i--)
			{
				System.out.println("print statement");
			}
		}
	}
}
```

—----------

2. While loop

A while loop is a control flow statement that allows code to be executed repeatedly based on a given Boolean condition. The while loop can be thought of as a repeating if statement.

Syntax >>

```
Initialization
while ( condition)
{
	loop statements…
	Increment / decrement
}
```

E.x.

```
class whileLoop
{
   public static void main(String args[])
   {
      int x = 1;
      while (x <= 4)                    // Exit when x becomes greater than 4
      {
         System.out.println(x);
         x++;
      }
   }
}
```

—----------

3. Do while

do while loop is similar to while loop with only difference that it checks for condition after executing the statements, and therefore is also called Exit **Control Loop.**

Syntax >>

```
do
{
        statements..
}
while (condition);
```

E.x

```
class dowhileloopDemo
{
   public static void main(String args[])
   {
      int x = 21;
      do
      {
         System.out.println(x);
         x++;
      }
      while (x < 20);
   }
}
```

—----------

Difference between while and do while loop

- “While loop” checks the condition first and then executes the statement(s), whereas “do while loop” will execute the statement(s) at least once, then the condition is checked.
- “While loop” is entry controlled loop whereas “do while loop” is exit controlled loop.
- In the “while loop”, we do not need to add a semicolon at the end of a while condition but we need to add a semicolon at the end of the while condition in the “do while loop”.
- “While loop” statement(s) is executed zero times if the condition is false whereas “do while loop” statement is executed at least once.
- “While loop” allows initialization of counter variables before starting the body of a loop whereas “do while loop” allows initialization of counter variables before and after starting the body of a loop.

==========