# Java Programming
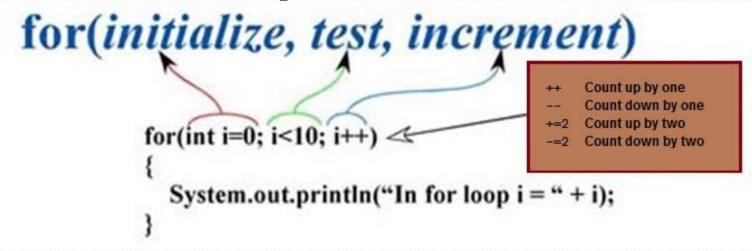
Loops

# What are loops and how do you write Java code?

- Frequently in a program you have to do something over and over. Java provides some convenient programming statements to let you do just that. They are called **loops**.

- **Four Types of Loops:**

  - `for`
  - `while`
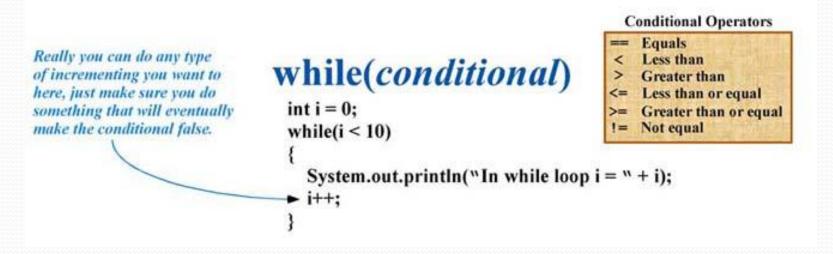  - `do...while`
  - `for...each`

# The *for* Loop

- In this type of loop you create a **loop counter** variable and initialize it to a starting value, you provide a test to determine when the loop is finished, and you provide a way of changing the value of the loop counter (either incrementing or decrementing). Everything that you want to do over and over is placed inside of braces.

for(*initialize, test, increment*)

```
for(int i=0; i<10; i++)
{
    System.out.println("In for loop i = " + i);
}
```

| | |
|---|---|
| ++ | Count up by one |
| -- | Count down by one |
| +=2 | Count up by two |
| -=2 | Count down by two |

# The *while* Loop

- In this type of loop you create a **loop counter** variable and initialize it before the loops is set up. The **while** line in the loop just contains the "test" conditional statement. As long as this statement is **true** the loop will continue. Something done inside the loop must eventually change the loop counter so that the conditional is false.

*Really you can do any type of incrementing you want to here, just make sure you do something that will eventually make the conditional false.*

```
while(conditional)
int i = 0;
while(i < 10)
{
    System.out.println("In while loop i = " + i);
    i++;
}
```

**Conditional Operators**

| | |
|---|---|
| == | Equals |
| < | Less than |
| > | Greater than |
| <= | Less than or equal |
| >= | Greater than or equal |
| != | Not equal |

# The *do...while* Loop

- In this type of loop you create a **loop counter** variable and initialize it before the loops is set up just as you do in the **while** loop. The main difference is that in the **do...while** loop the conditional is not checked till the end of the loop where as in the **while** loop it is checked at the beginning. That means that if a **while** loop's conditional is false to start with it will not execute at all. But, a **do...while** loop will always execute at least once because even if its' loop counter is false to start with, that will not be checked till the end of the loop.

**do...while(*conditional*)**

```
int i = 0;
do
{
    System.out.println("In do-while loop i = " + i);
    i++;
}
while(i<10)
```

# The *for...each* Loop

- A **for...each** construct was introduced in JDK 1.5.0.
- It is referred to as the "**Enhanced for Loop**", the "**For-Each Loop**", and the "**foreach statement**".

**for...each (*element, array*)**

```
String[] fruits = new String[]{"Orange", "Apple", "Banana"};

for (String fruit : fruits) {
    System.out.println(fruit);
}
```