

# Loops in Java

# Intro to Java Loops

- A programming loop is one that forces the program to go back up again.
- If it is forced back up again you can execute lines of code repeatedly.
- As an example, suppose you wanted to add up the numbers 1 to 10. You could do it quite easily in Java like this:
- **int addition = 1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10;**
- This method is only good if you want to add few numbers not if you needed to add up the numbers 1 to a 1000.
- Instead, you can use a loop to go over a line of code repeatedly until you've reached 1000.

# For Loops

- Let's look at how a for loop looks like and how to set up one.

```
for (int x = 0; x < 5; x++) {  
}
```

- First, let's examine the ++ operator using a simple integer variable:

```
int m = 5;
```

- Here **m** is equal to 5. Adding the following `m++` to **m** adds one to **m** without having to do `m = m + 1;`
- In this example `m` is equal to 5, so this new line makes **m** equal to 6. Likewise `m--` subtracts one so this new line makes **m** equal to 4.
- The loop starts off with the "**for**" keyword in lower case.
- Next the terms of the loop go inside the parenthesis `( )`.

# For Loops

- The first thing you see is **int x = 0**; this is the starting point of the loop.
- It says that at the beginning of the loop, an integer variable named x is going to be zero.
- Next you will see **x < 5**; This is saying that the loop should continue so long as x is less than 5. The last part says **x++**. This means that every time the loop repeats, it will add one to **x**.
- In our example we're counting up by one, but we could also add 2 to **x** every time the loop starts over:

```
for (int x = 0; x < 5; x = x + 2) {  
  
}
```

- Let's say we wanted our program to print out Hello world! 5 times. This is how we would use a for loop to do it:

```
for (int x = 0; x < 5; x++) {  
    System.out.println("Hello world!");  
}
```

# For Loop Example

- `public class ForLoop {`
- `public static void main(String[] args) {`
- `// TODO Auto-generated method stub`
- `int loopVal;`
- `int end_value = 11;`
- `for (loopVal = 0; loopVal < end_value; loopVal++) {`
- `System.out.println("Loop Value = " + loopVal);`
- `}`
- `}`
- `}`

# While Loop

- The second type of loop in Java is the "while loop".
- A while loop is just a conditional that repeats itself as long as the condition stays true.

```
while (num == 3) {  
  
}
```

- A while loop looks just like an If Statement; just replace the "if" keyword with the keyword "while".
- The statement above says that while num is equal to 3, keep executing whatever code is inside the brackets.
- When the bottom of the code in the brackets is reached, it goes back up to the top, checks to see if num is still equal to 3, and then repeats again.

# While Loop

- It is up to the coder to insure that this loop eventually has a way to stop. Java will NOT figure this out for you. If you don't you will be stuck in an infinite loop.
- An infinite loop example:

```
while (num == 3) {  
    num = 3;  
}
```