Computer Programming Using JAVA
COP 2800 - Fall 2016

Lecture 7: Java Loops

By Ahmed E. Khaled
Recap to previous lecture!

• How to define method in Java?
• Why we need methods?
• What is a nested method?
Lecture 7 : Java Loops

- Continue Java Methods
- Java Loops.
public static int sumBoth ( int num1 , int num2 ){

    int result = num1 + num2;
    return result;
}

public static void main(String[] args) {

    int x = 100;
    int y = 200;
    int z = sumBoth ( x , y );

    System.out.println("result = " + z );

    int m = 300;
    int n = 500;

    System.out.println("result = " + sumBoth ( m , n ) );
}
What is Actually Passed to a Method?

- In Java, parameters sent to methods are passed-by-value.

- When a method is called, a *copy of the value* of each argument is passed to the method.

- This copy can be changed inside the method, but such a change will have *NO effect* on the actual variable!

- The actual variable is not affected, regardless of the changes made to the parameter inside the method.
public static void incrementValue(int num) {
    System.out.println("User Passed value " + num);
    num = num + 2;
    System.out.println("the function updated the value to " + num);
}

public static void main(String[] args) {
    int x = 5;
    System.out.println("Value of x is " + x);
    incrementValue(x);
    System.out.println("Value of x is " + x);
    x = 12;
    incrementValue(x);
    System.out.println("Value of x is " + x);
}
run:
Value of x is 5
User Passed value 5
the function updated the value to 7
Value of x is 5
User Passed value 12
the function updated the value to 14
Value of x is 12
BUILD SUCCESSFUL (total time: 13 seconds)
public class WelcomeJava {

    public static int num1 = 10;

    public static void changeGlobal(){
        int num3 = 30;
        System.out.println( num3 );
        System.out.println( num1 );

        num1 = 100;
    }

    public static void main(String[] args) {
        int num2 = 20;
        System.out.println( num2 );

        System.out.println( num1 );
        changeGlobal();
        System.out.println( num1 );
    }
}
Control Structures

• As discussed before, Java provides six control structures (of two types):
  
  • Branches : Choose among two or more possible courses of action (e.g. If / If-Else / Switch)
  
  • Loops : Repeat a sequence of statements over and over. (e.g. While / Do-While / For)
Java Loops

• How to display “Hello Java!” hundred times?
  System.out.println("Hello Java!");
  System.out.println("Hello Java!");
  ....
  System.out.println("Hello Java!");
  System.out.println("Hello Java!");
  System.out.println("Hello Java!");

• Java loop controls how many times an operation or a sequence of operations is performed in succession.
  loop for 100 times {
    System.out.println("Hello Java!");
  }
Java Loops

• Repeat a given set of statements over and over, of course not forever (*infinite loop*).

• A one-time execution of a loop body is referred to as an *iteration* (or repetition) of the loop.

• During this course, we will check different types of java Loops:
  1. While loop.
  2. For loop.
  3. Do while.
  4. Nested Loops
The While Loop

• While loop will repeat a statement over and over, but only so long as a specified condition remains true. A while loop has the form:

```java
while ( boolean Expression ) {
    Instruction
    Instruction
    ....
    Instruction
}
```
• When the compiler comes to a while statement, it evaluates this boolean expression.

• If false, the computer skips the while loop and proceeds to what is after.

• If true, the computer executes block of statements inside the loop. Then returns to the beginning of the loop and repeats the process of re-evaluating the boolean expression.

• This will continue over and over until the value of the expression is false.
Examples

```java
int count = 0;
while (count < 5) {
    System.out.println("Hello Java!");
    count = count + 1;
}
```

*Note: Count = count +1 is the same if I said Count++*
• Here is an example of a while loop that simply prints out the numbers 1, 2, 3, 4, 5:

```java
int number = 1;
while ( number < 6 ) {
    System.out.println(number);
    number = number + 1; // Go on to the next number.
}
System.out.println("Done!");
```
Here is an example of a while loop that simply prints out the numbers 6, 5, 4, 3, 2:

```java
int number = 6;
while ( number > 1 ) {
    System.out.println(number);
    number = number - 1; // Go on to the prev number.
}
System.out.println("Done!");
```
```java
int number = 1;
int Sum = 0;
while ( number < 6 ) {
    Sum = Sum + number;
    number = number + 1;
}
System.out.println("Done!" + Sum);
```
• What happens if the loop is written as follows?

```java
int sum = 0,
Int i = 1;
while (i < 10) {
    sum = sum + i;
}
```
System.out.println("Enter a positive integer greater than 5 ");
int inputNumber;
inputNumber = in.nextInt();
int count = 0;
while (inputNumber > 5) {
    count = count + 1;
    System.out.println("Enter a positive integer greater than 5 ");
    inputNumber = in.nextInt();
}
System.out.println("You entered " + count + " correct values");