COMP 110
Introduction to Programming

Fall 2015
Time: TR 9:30 – 10:45
Room: AR 121 (Hanes Art Center)

Jay Aikat
FB 314, aikat@cs.unc.edu

Previous Class

- What did we discuss?
Today

- Announcements
  - Assignment 2 : Due Friday, Oct 2 @ 11:55 PM
    http://comp110.com/assignments/the-worried-wizard

- Poll everywhere
  http://help.unc.edu/help/poll-everywhere-faq/

- More For Loops

Local Variables

- Open Eclipse
- New Java project etc.. You know the drill!

```java
public class test123 {
    public static void main(String[] args) {
        int num1 = 5;
        int count;
        for (count = 0; count <= num1; count++){
            System.out.println(count);
        }
    }
}
```
Local Variables

```java
public class test123 {
    public static void main(String[] args) {
        int num1 = 5;
        int count;
        for (count = 0; count <= num1; count++) {
            int num2 = 10;
            System.out.println(count);
            System.out.println(num2);
        }
    }
}
```

Local Variables

```java
public class test123 {
    public static void main(String[] args) {
        int num1 = 5;
        int count;
        for (count = 0; count <= num1; count++) {
            int num2 = 10;
            System.out.println(count);
            System.out.println(num2);
        }
        System.out.println(num2);
    }
}
```
import java.util.*;
public class Example1 {
    public static void main(String[] args) {

        String str = "Yes";
        int count = 0;
        Scanner keyboard = new Scanner(System.in);

        while (str.equalsIgnoreCase("Yes")){
            System.out.println(count);
            count++;
            System.out.println("Would you like to continue?");
            str = keyboard.next();
        }
    }
}

Debugger in Eclipse

- **Debugging** allows you to run a program interactively while watching the source code and the variables during the execution
- Using **breakpoints** in the source code, you specify where the execution of the program should stop
- Add line numbers
- Toggle breakpoint where needed

https://www.youtube.com/watch?v=dHYM3b3ZEjU
Debugger in Eclipse – exercise1

```java
int count = 0;
int number = 5;
while (count <= number) {
    System.out.println(count);
    count++;
}
```

Debugger in Eclipse – exercise2

```java
int num = 5;
for (int i = 0; i <= num; i++) {
    System.out.println(i);
}
```
int sum = 0;
int numToSum = 0;
Scanner keyboard = new Scanner(System.in);
System.out.println("Please enter a number");
numToSum = keyboard.nextInt();
for (int i=0; i <= numToSum; i++){
    sum += i;
}
System.out.println("The sum is " + sum);
keyboard.close();

Next class

• More on loops