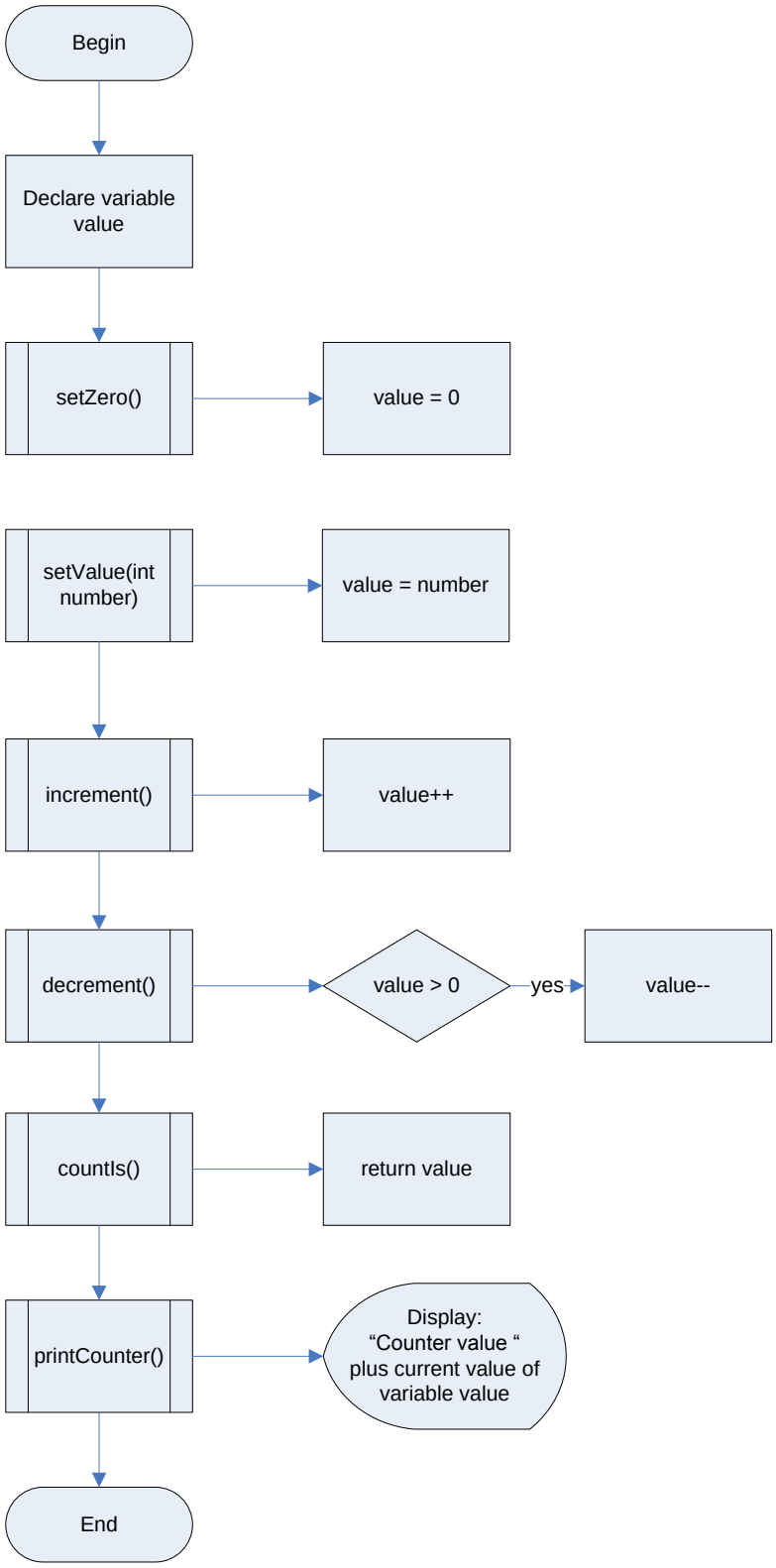
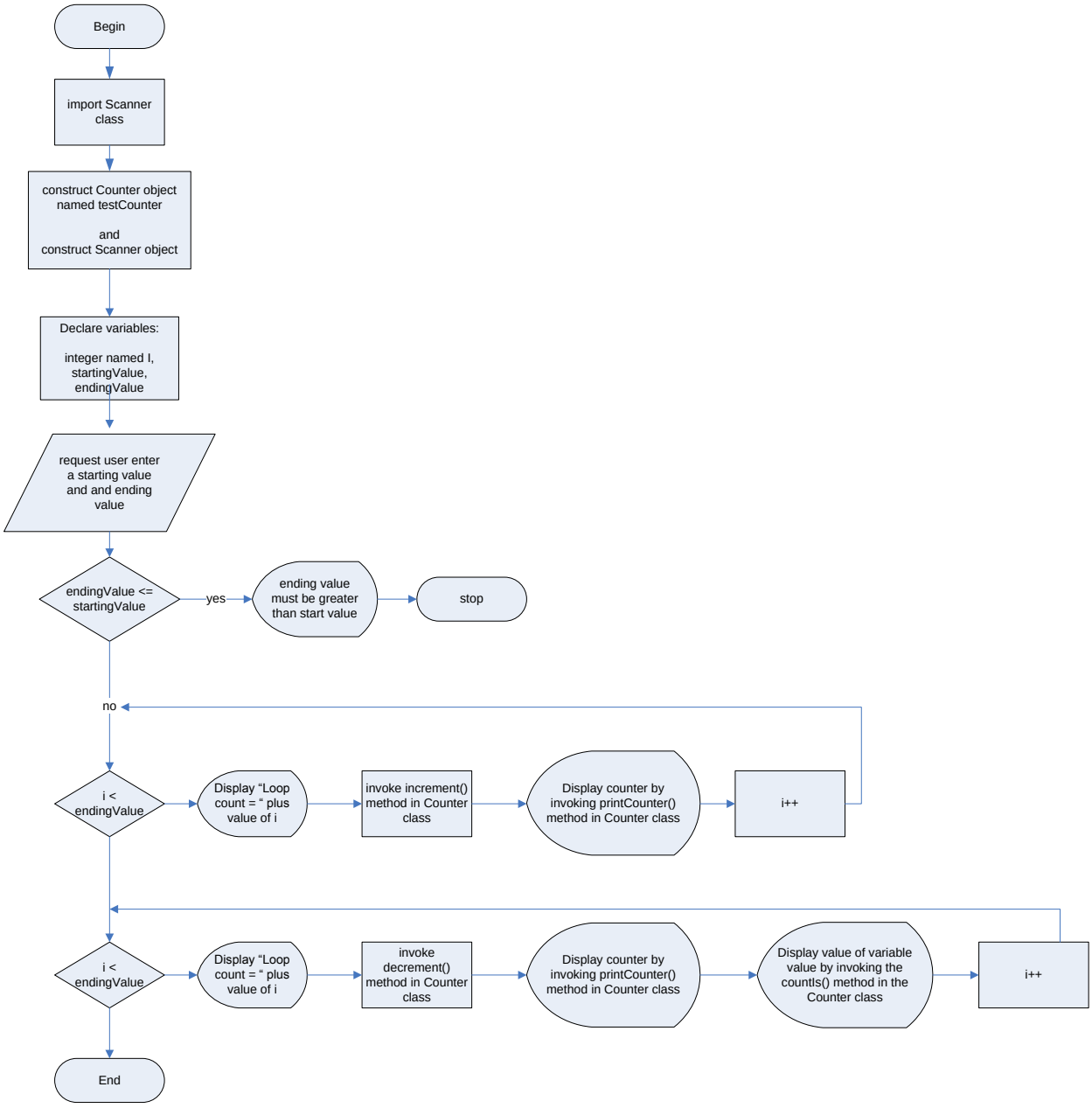
CIT 149: Java I
Chapter 5 Lab 4 Assignment
Debugging

In this lab we will create the programming project # 2 on page 361. This is a debugging assignment and I have purposely made errors for you to discover and correct. This assignment contains two programs. The external program contains methods that will be used by the driver program, but will not contain the main method. There are two flowcharts for this assignment. The first flowchart for the Counter program is:



The second flowchart, for the CounterTest program, is:



1. Open a new document in TextPad and save the program as Counter.java.
2. Type the following code that will create different methods. Each of these methods will set different values for the variable *value*:
    - the first method setZero() will set the value to zero
    - the second method setValue() will set the value to the value passed to the method
    - the third method increment() will increase its value by 1
    - the fourth method decrement() will decrease its value by 1 if it is already greater than 0
    - the fifth method will return the current value of the variable

- the last method will print out the current value.

The code is:

```
public class Counter
{
    private int value

    public void setZero()
    {
        value = 0;
    }

    public setValue(int number)

    {

         value = number;

    }

    public void increment()
    {
        value++;

    public void decrement()
    {

        if(value > 0);
            value--;
    }

    public int countIs()
    {

        return value;
    }

    public void printCounter()
    {
        System.out.println("\nCounter value = " + value); //insert a blank line before this
    }
}
```

3. Compile the program and fix the errors until it compiles correctly.

4. Our second program will test the methods from the first program.
5. Open a new document in Textpad and save the program as CounterTest.java.
6. First we must import the Scanner class by typing:

```
import java.util.Scanner;
```

7. We type our class header, opening brace, main method header and opening brace.

```
public class CounterTest
{

 public static void main(Strings[])
 {
```

8. Next we constructor and/or declare objects and variables. Type:

```
    Scanner keyboard = new Scanner(System.in);
   Counter testCounter = new Counter();

   int startingValue, endingValue;
```

9. We request user input and display the value as the counter. Type:

```
  System.out.println("Change the starting value by entering a number");
startingValue = keyboard.nextInt();

System.out.println("Set the ending value by entering a number.  This number MUST be higher
than the starting value");
endingValue = keyboard.nextInt();

testCounter.setValue(startingValue);
System.out.println("After setValue() testCounter = " + testCounter.countIs());
```

10. We include an if statement in case the user enters an ending value that is less than or equal to the starting value. If so a message displays and the program ends. Type:

```
if(endingValue <= startingValue)
{
    System.out.println("The ending value must be higher than the starting value.  This program
will now end");
    System.exit(0);
```

```
}
```

11. Our first for loop is used to test the increment method. Type:

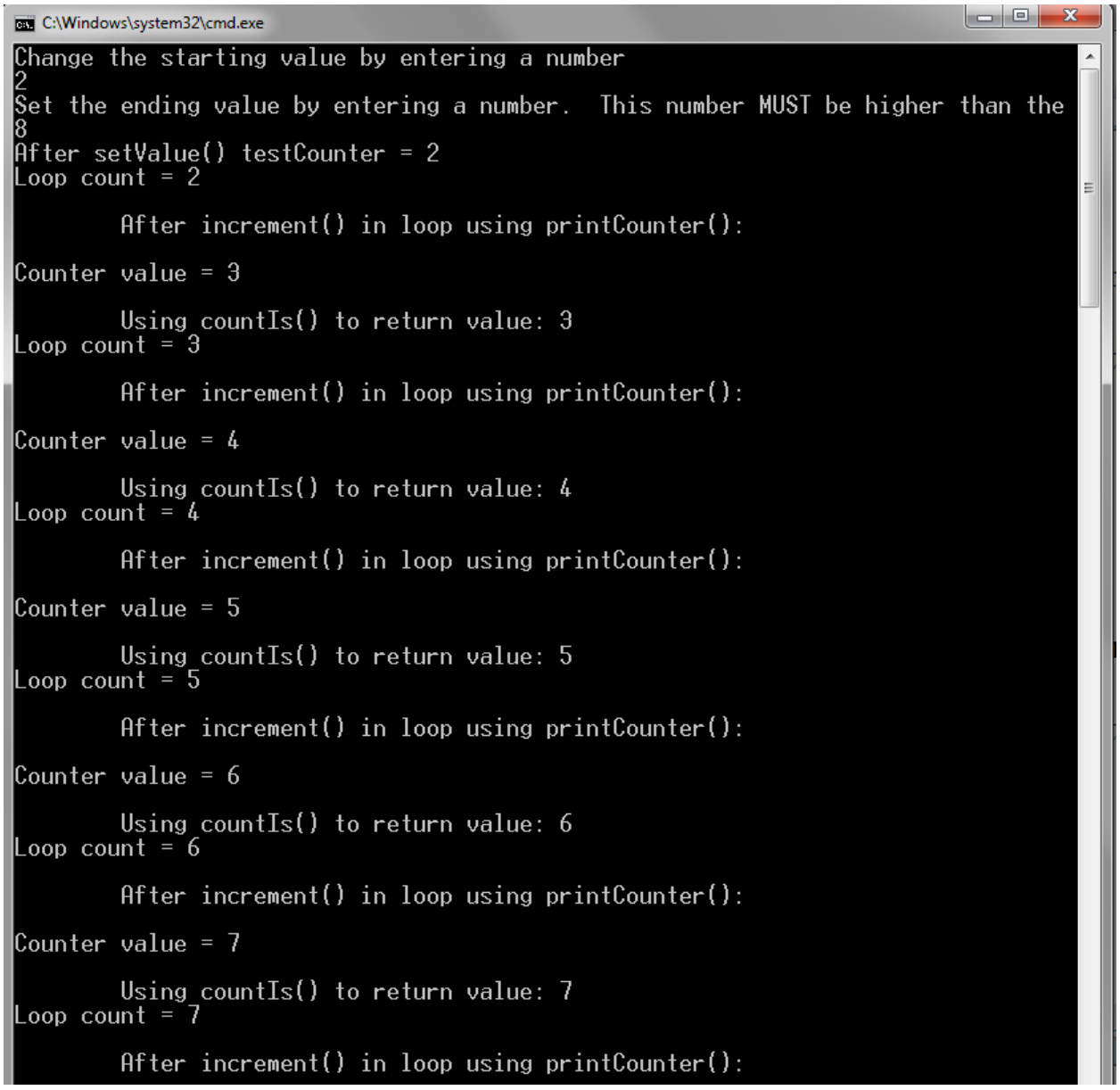
```
 for(i=startingValue; i < endingValue; i++)
{
    System.out.println("Loop count = " + i);
    testCounter.increment();
    System.out.println("\n\tAfter increment() in loop "+ "using printCounter(): ");
    testCounter.printCounter();
    System.out.println("\n\tUsing countIs() to return value: " + testCounter.countIs());
}
```

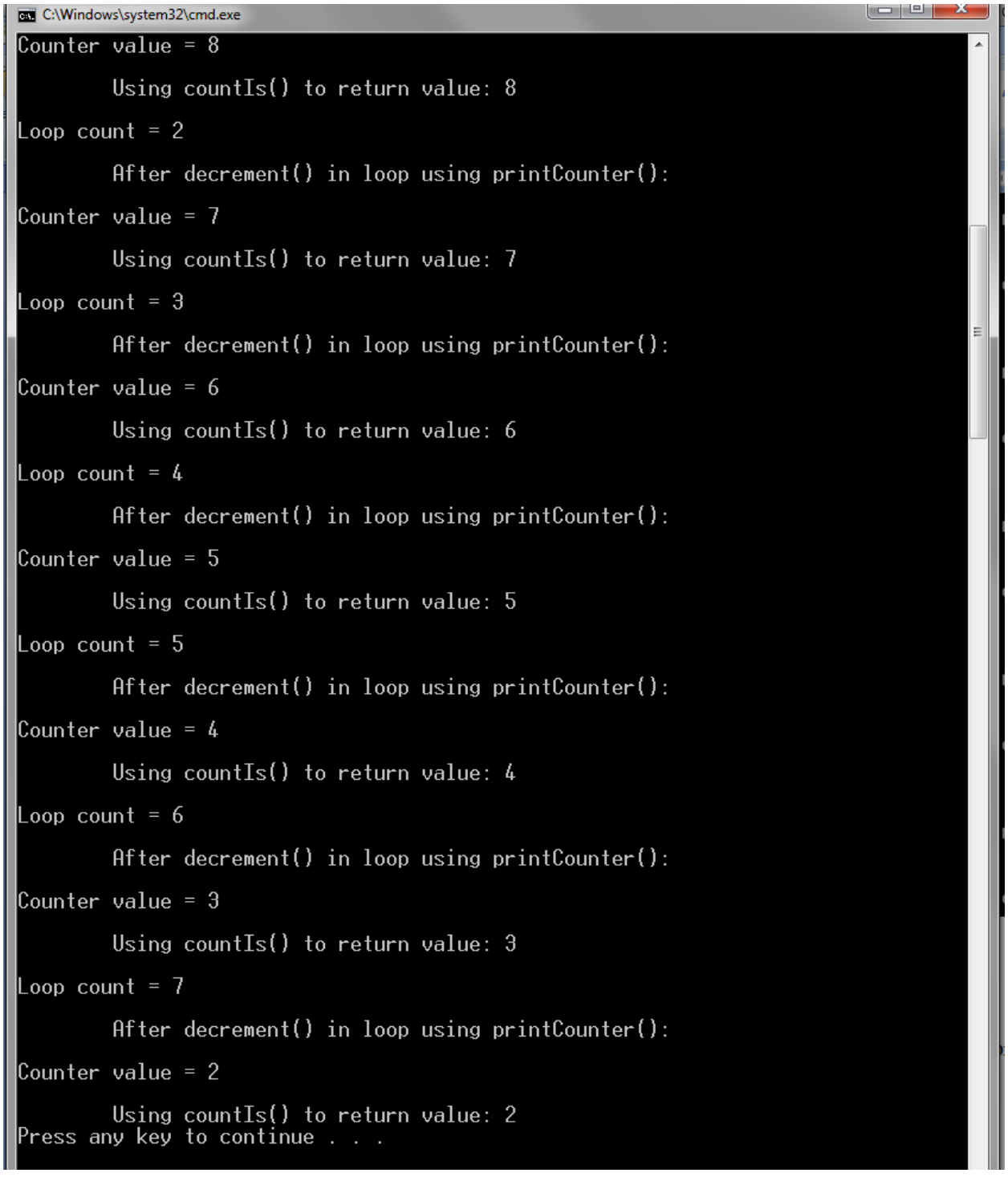
12. We also want to test the decrement() method. Type:

```
   for(i=startingValue; i < endingValue; i++)

   {

       System.out.println("\nLoop count = " + i);

       testCounter.decrement();

       System.out.println("\n\tAfter decrement() in loop "  + "using printCounter(): ");

       testCounter.printCounter();

       System.out.println("\n\tUsing countIs() to return value: " + testCounter.countIs());

   }
```

13. You should be able to follow this program fairly easily. The first for loop prints out the current value of the variable *i* which is set within the argument of the loop. The method invokes the increment() method for testCounter and then displays text; invokes the printCounter() method.
14. Close the main method and class.
15. Compile the program and fix the errors.
16. Run the program. If you do not get the following results then there are still errors.

It will continue printing out this. The print is in two parts due to the size of the print out screen:


C:\Windows\system32\cmd.exe
Change the starting value by entering a number
2
Set the ending value by entering a number. This number MUST be higher than the
8
After setValue() testCounter = 2
Loop count = 2
After increment() in loop using printCounter():
Counter value = 3
Using countIs() to return value: 3
Loop count = 3
After increment() in loop using printCounter():
Counter value = 4
Using countIs() to return value: 4
Loop count = 4
After increment() in loop using printCounter():
Counter value = 5
Using countIs() to return value: 5
Loop count = 5
After increment() in loop using printCounter():
Counter value = 6
Using countIs() to return value: 6
Loop count = 6
After increment() in loop using printCounter():
Counter value = 7
Using countIs() to return value: 7
Loop count = 7
After increment() in loop using printCounter():



17. In the above figure I entered 2 as the starting value and 8 as the ending value.
18. After fixing all the errors compress the following files into a single zip or rar file and submit to the appropriate drop box.

Counter.java
Counter.class
CounterTest.java
CounterTest.class