

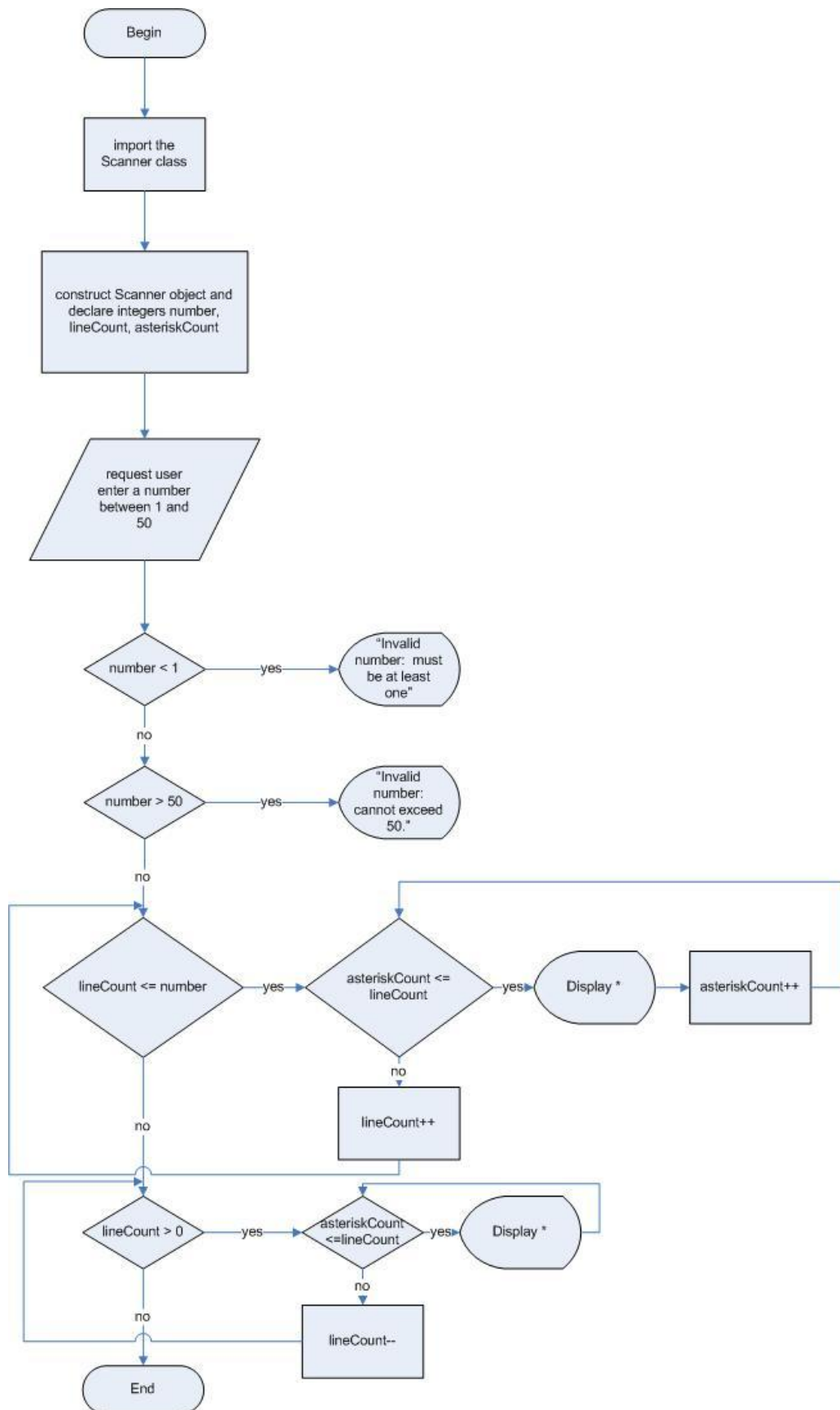
CIT 149: Java I

Chapter 4 Lab 3

In this lab we will use the for loop to display a series of asterisks in the shape of a triangle. When the program is run it will display as:

[illegible]

In this case I typed in 25 which designate the maximum number of asterisks in the triangle's widest line. The flowchart for this assignment is:



1. Open a new document window in TextPad and save the program as TriangleOfAsterisks.java.
2. Type block comments that include your name, the date, and the purpose of the program.
3. Type the class header, opening brace, main method header, and the main method's opening brace.
4. Within the main method construct a Scanner object named keyboard.
5. Declare integers of number, lineCount, and asteriskCount.
6. Type the following code which will request the user to enter a number between 1 and 50, and will assign the input to the variable *number*.

```
System.out.println("Enter an integer from 1 to 50: ");
number = keyboard.nextInt();
System.out.println();
```

Notice that it also prints a blank line? This is for formatting purposes only.

7. Our next step is to check to make certain the user entered a valid number. This is done using if statements. Type:

```
if(number < 1)
    System.out.println("Invalid number: must be at least one.");
else if(number > 50)
    System.out.println("Invalid number: cannot exceed 50.");
```

8. If the number is not a valid number then we need to create the code to display the asterisks. All the code is enclosed in an else statement since we had an if, and an else if. Start off by typing:

```
else
{
```

9. First we will print out the first half of our triangle of asterisks using a for loop with a nested for loop. Type:

```
for(lineCount = 1; lineCount <= number; lineCount++)
{
    for(asteriskCount = 1; asteriskCount <= lineCount; asteriskCount++)
    {
        System.out.print("*");
    }
    System.out.println();
} // end first half for
```

- When nesting for loop the inner loop is run first. In this case the first time only a single asterisk will be displayed. Then the outer for loop is checked and 1 is added to *lineCount*, the inner for loop is run again, having increased *asteriskCount* by 1 after running the loop the previous time. Since *asteriskCount* now equals 2, 2 asterisks will be displayed, and so on....
10. Now we have to print out the bottom half of our triangle. Again for loops are used. Type:

```
for(lineCount = number -1; lineCount > 0; lineCount--)
{
    for(asteriskCount = 1; asteriskCount <= lineCount; asteriskCount++)
    {
        System.out.print("*");
    }
    System.out.println();
} // end second half for
```

- This is also a loop within a loop and acts the same way as the previous ones except in reverse.
 - The nested for loop is the same as the previous nested loop. The difference is in the outer for loop.
 - This loop sets the starting point of *number* minus 1, the loop will run as long as *lineCount* is greater than 0. Notice the increment though? *lineCount* will decrease by one every time the loop is run. The variable *lineCount* was set with the previous for loop. So if you had set the number of lines to 25 then *lineCount* would end with a value of 26 after the previous for loop since its value started with 1.
11. To finish the program close the else statement, main method and the class.
 12. Compile the program and fix any errors if necessary.
 13. Run the program and see the results you get.
 14. Compress the TriangleOfAstericks.java and .class file into a single zip or rar file and submit the appropriate drop box.