South Louisiana Community College
ASDV 1220, Programming Fundamentals

Learning Objectives

After completion of this lab, you should be able to

1. Understand Debugging and the Debugger of Netbeans

Create project Lab13LastName

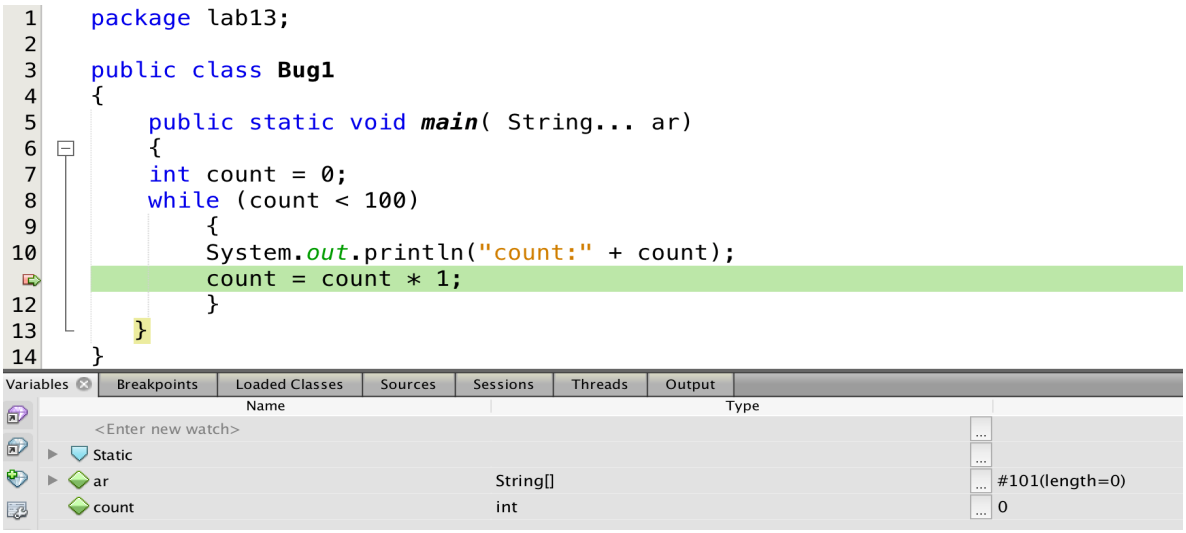
Problem 1

Create a class **Bug1**. Follow the steps 1 though 4 below and fix the bug (infinite loop):

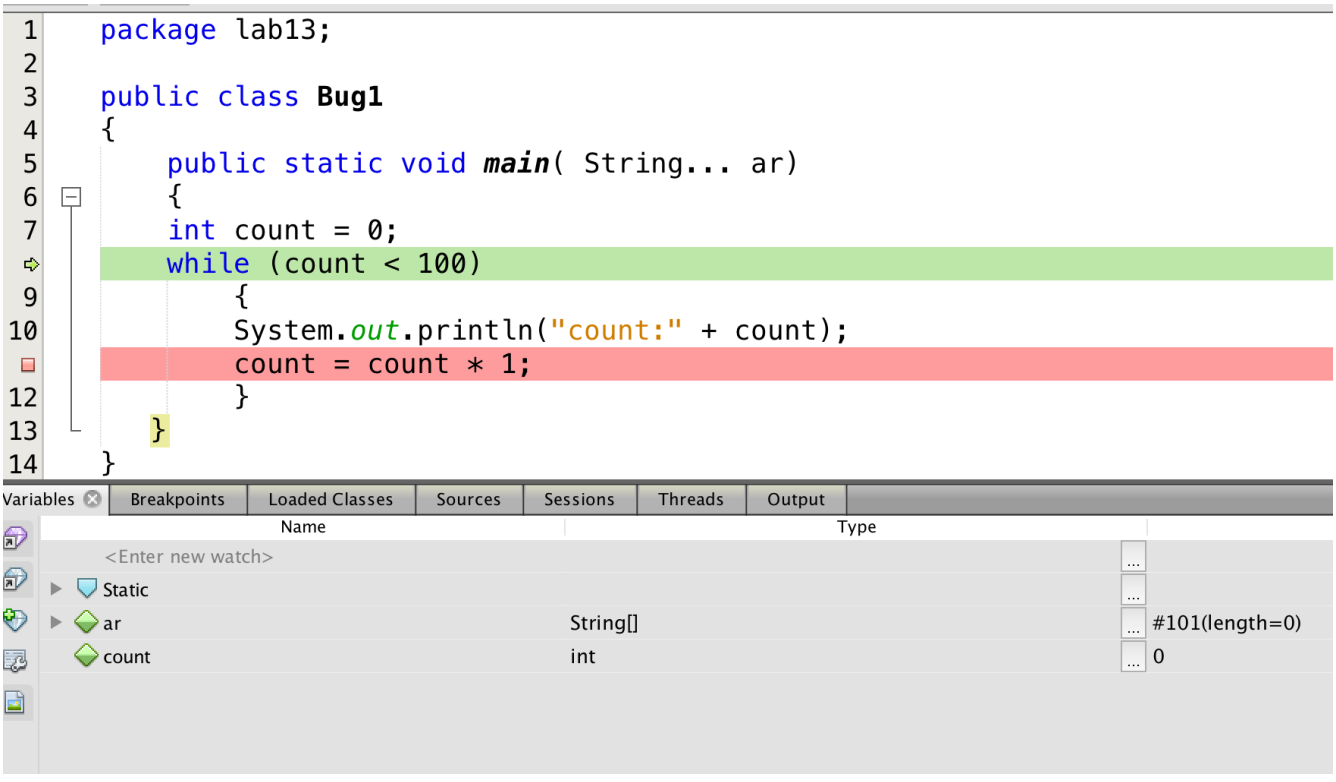
1. **Set a line breakpoint at line 11** by literally clicking on number 11. This will pause the program's execution at the line 11 while running in Debug Mode.
2. To run the program in Debug Mode, right click on Bug1.java, under folder lab13, and select **Debug File**.

```
1   package lab13;
2
3   public class Bug1
4   {
5       public static void main( String... ar)
6       {
7       int count = 0;
8       while (count < 100)
9           {
10          System.out.println("count:" + count);
            count = count * 1;
12          }
13      }
14  }
```

3. The program will stop at line 11 as shown below:
   Note, that the green color on line 13 indicates that line 13 is the NEXT line for execution.



4. Press **F8**, or from the Debug menu, click **Step Over Expression**. The F8 will take you to next line of execution which is line 8 as shown below. Observe the contents of variable **count** in the variables-window. If the variables window is not visible click Window>Debugging>Variables



5. After you fix the bug, the program displays count 0 to count 99.

Problem 2

Create a class **Bug2**. The program Bug2 converts a positive decimal number into binary number and displays it. However the binary displayed is incorrect. Read the commented algorithm inside the code, and type in the code. **There are 2 bugs in** the code. (2 lines have to be corrected). Use the Debugger and fix the 2 bugs by setting breakpoints and by using F8 to examine the contents of variables at each line. Try entering *number **123** as input* and go though the loop, line by line. If you don't find the problems(bugs) immediately try again, or try different input numbers. Try setting breakpoints at lines 28, 37, 39.

```
1   package lab13;
2   import java.util.Scanner;
3   public class Bug2
4   {
5   public static void main(String[] args)
6   {
7               Scanner in = new Scanner(System.in);
8               int decimalNumber;
9               String binaryNumber;
10              System.out.print("Enter a positive integer: ");
11              decimalNumber = in.nextInt();
12              int x = 1;
13              if (decimalNumber <= 0)
14                    System.out.println("ERROR: entered integer is nonpositive.");
15              else {
16                    binaryNumber = "";
17                    // algorithm step by step
18                    // initial: binaryNumber = "", decimalNumber = 123
19                    // step 1 : binaryNumber = "1 ", decimalNumber = 61
20                    // step 2 : binaryNumber = "11 ", decimalNumber = 30
21                    // step 3 : binaryNumber = "011 ", decimalNumber = 15
22                    // step 4 : binaryNumber = "1011 ", decimalNumber = 7
23                    // step 5 : binaryNumber = "1 1011 ", decimalNumber = 3
24                    // step 6 : binaryNumber = "11 1011 ", decimalNumber = 1
25                    // step 6 : binaryNumber = "111 1011 ", decimalNumber = 0
26                    // stop   : (decimalNumber != x)
27
28                    while (decimalNumber != x)
29                          {
30                              //> add spaces to separate 4-digit groups
31                          if (binaryNumber.length() % 5 == 0)
32                               binaryNumber = " " + binaryNumber;
33                              //> extract last digit in binary representation
34                              // and add it to binaryNumber
35                          binaryNumber = (decimalNumber % 2) + binaryNumber;
36                              //> cut last digit in binary representation
37                          decimalNumber /= 3;
38                    }
39                    System.out.println("Binary: " + binaryNumber);
40              }
41        }
42    }
43
```