Tutorial 3

1. Convert the while loop example above into a functionally equivalent do while loop.

```
public static void main(String[] args) {
// add up the numbers
int count = 0;
while (count < 9)
 {
    System.out.print(" Hello ");
    count = count + 1;
 }
System.out.println();
}
```

2. Write a Java program to accept two integers from the user to display all the numbers in between. Modify the code to make sure the first number that the user enters is always lower than the second.

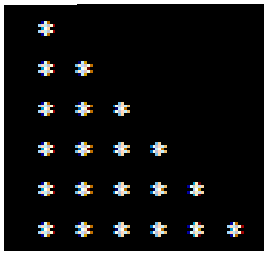
3. Write a java program using a while loop to accept an positive integer n and a letter from the user to print the letter *n* times.

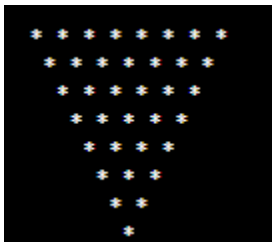
```
Enter a number : 5
Enter a letter : Y

output : YYYYY
```

4. The Fibonacci series is a follows:
   1, 1, 2, 3, 5, 8, 13, 21, 34

   Can you see how each number is generated? After the first two numbers, each number is the sum of the previous two numbers. This series is unusual in that it can be used to describe natural phenomena, such as angular displacement of plant leaves for maximizing incident light.
   Write a program to compute this series to the n_{th} term.

5. Draw the following patterns using for loops





6. See the program below.
   Number to the power '*power'* is number * number * number ….. (*power* number of times)
   This program multiplies number by number while the counter is less than power.
   Why doesn't it work? How do we debug?

```
import java.util.*;
import java.io.*;
public class Main {
public static void main(String[] args)
{

      int number, power, count;
      int total = 0;
      Scanner input = new Scanner(System.in);
      System.out.println("Enter number : ");
      number = input.nextInt();
      System.out.println("Enter power ");
      number = input.nextInt();
      count = 0;
      while (count <= power )
      {
            total = number * number;
      }
      System.out.println(" the answer is " + total);
      }
}
```

7. Factorial 5 (5!) = 5x4x3x2x1 . The program below asks for the factorial number. Dry run it. Try entering in a 4 for the number. Why doesn't the loop work? Put a print statement to see how many times the loop goes around.

```
import java.util.*;
public class Main
{
      public static void main(String[] args) {
      int number, count, factorial;
      int total = 0;
      Scanner input = new Scanner(System.in);
      System.out.println("Enter number ");
      number = input.nextInt();
      count = 0;
      while (count > number )
      {
            number = count;
            factorial = total * number;
      }
      System.out.println("the factorial is " + total);
   }
}
```

   fix the code so that it prints the factorial.

8. Given any number, we can create a new number by **adding the sums of squares of digits of that number**. For example, given 203, our new number is 4 + 0 + 9 = 13. If we repeat this process, we get a sequence of numbers:

   n -> 13 -> 10 -> 1 -> 1

   Sometimes, like with 203, the sequence reaches (and stays at) 1. Numbers like this are called happy.Not all numbers are happy. If we started with 11, the sequence would be:

   11 -> 2 -> 4 -> 16 -> .

   This sequence will never reach 1, and so the number 11 is called **unhappy**.