Home Work 4-2 – Recursion

A recursive method is a method that calls itself either directly or indirectly through another method. Recursion is discussed in an example on the course web page using the Triangle class.

Factorial is a popular mathematical operation. It is defined as follows:

The factorial of a non-negative integer *"n"*, written " $n!$ "(and pronounced "n factorial"), is defined as the product:

$$n! = n \times (n - 1) \times (n - 2) \times (n - 3) \ldots \times 1$$

For example:

$$5! = 5 \times 4 \times 3 \times 2 \times 1$$ which is equal to 120.

There are two special cases: $1! = 1$ and $0! = 1$.

Back in unit 1 you saw the following program. It calculates the factorial of any number using a simple loop (i.e. non-recursively).

```
import javax.swing.*;

public class Factorial
{
   public static long factorial (long number)
   {
         long fact = 1;
         for (long count = number ; count >= 1 ; count--)
         {
               fact = fact * count;
         }
         return fact;
   }

   public static void main (String [] args)
   {
         String input = JOptionPane.showInputDialog("Enter a non-negative integer: ");
         long num = Long.parseLong(input);

         JOptionPane.showMessageDialog ( null,num + "! = " + factorial(num),
                                  "Factorial",JOptionPane.PLAIN_MESSAGE);
         System.exit(1);
   }
}
```

Rewrite the above program using a recursive method.

Notice that given the above formula for factorial it can be expressed as follows:

$$n! = n \times (n - 1)!$$

Also note that the special case for the recursive method is that 0! = 1 and 1! = 1. Therefore the recursive factorial method can be started as follows:

```
public long factorial (long number)
{
     // special case code
     if (number <= 1)
     {
          return 1;
     }
          ...
}
```

Print out the new program and submit it.