Java Programming

Assignment 06

Create a project called **Assignment06**. Create a Java class, also called Assignment06, and copy this code into your IDE:

Assignment06.java

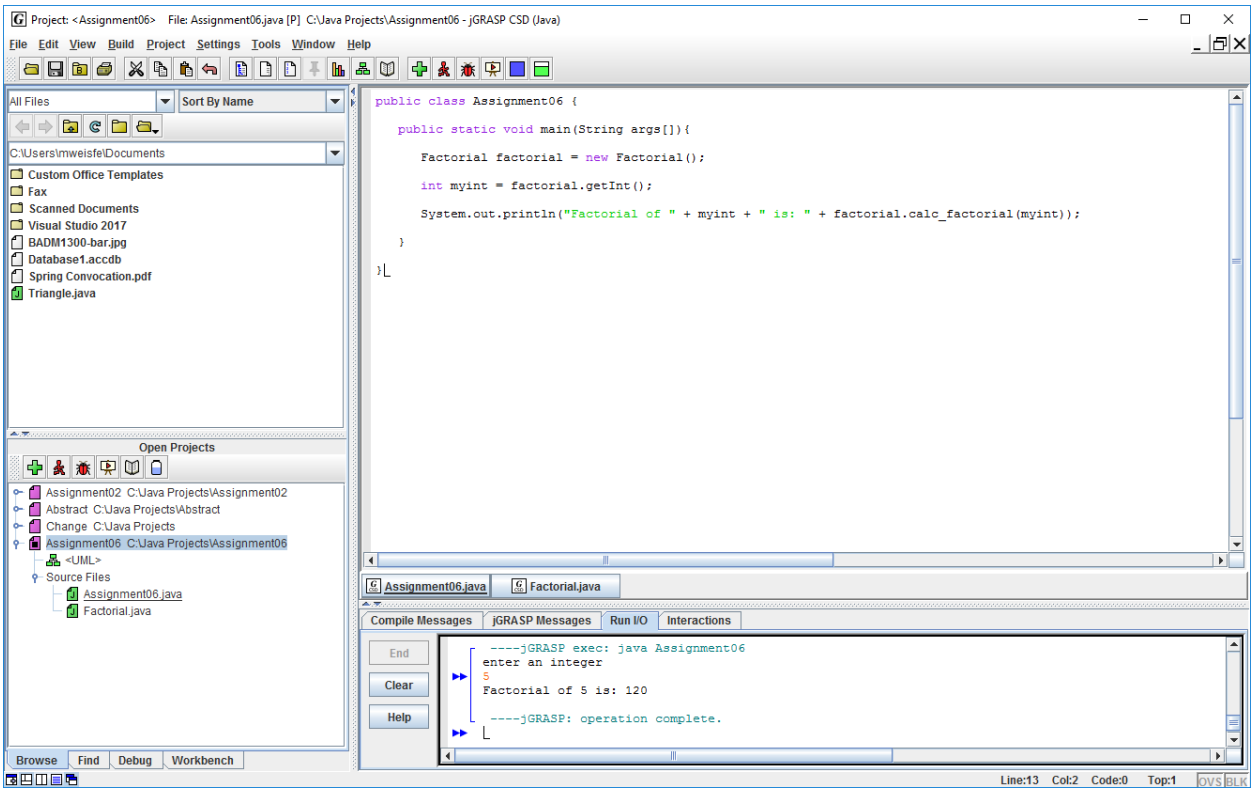
```
public class Assignment06 {

   public static void main(String args[]){

      // create a Factorial object
      Factorial factorial = new Factorial();

      // obtain an integer from the console
      int myint = factorial.getInt();

      // calculate the factorial of the number
      System.out.println("Factorial of " + myint + " is: " +
         factorial.calc_factorial(myint));

   }

}
```

Next, create a class called Factorial.java that will provide methods to obtain an integer from the console and also calculate the factorial of that number. *I have provided a hint on how to get the integer from the console by including the import line*.

Factorial.java

```
import java.util.Scanner;

public class Factorial {

      public int calc_factorial(int f) {

            // put method code here

      }

      public int getInt() {

            // put method code here

      }

}
```

In **jGrasp** (the IDE that I am using) the screen will look like this after a successful compile and execution.



This is a working application – when I enter 5 the answer 120 is calculated and displayed.

All that you need to do (and I don't say that in a trivial manner) is to 'flesh out' the 2 methods and execute a successful run of the main application that I provide.

What to Turn In

All you need to turn in is the completed Factorial class - in a file called Factorial.java.

So please upload the file Factorial.java to Blackboard. I don't need the entire project. I will simply add the file you submit to my project and test it.

Exploration (*Just for Fun* ☺)

Here are some extra things you can try (not required and not to be turned in).

Implementing the Fibonacci sequence with recursion

http://introcs.cs.princeton.edu/java/23recursion/Fibonacci.java.html

This is actually a pretty subtle and sophisticated concept to understand - it can be really interesting. Many years ago when I was a *young* software tester for Rockwell, I wrote a test engine that implemented a recursive algorithm. After many, many missteps I was pretty discouraged. When it finally worked (it took a week) I literally jumped out of my chair and I still remember how happy I was to really understand what it was doing. Any software developer who has experienced something similar will empathize ☺.