

A05

Due Date: Wednesday, March 1st

File(s) to be submitted: `MoreMath.java`

[Sample output](#)

Math Functions (Value-Returning Methods)

Summary

Create a class that works with my driver program (**A05**) to produce some mathematical output. Your class provides definitions for the undefined methods of that program.

Note: You will not be able to run the driver program until you have made a version of **MoreMath** *with every method defined*. (The definitions don't have to be *correct*, but they must be there, and they must return *something*. It's called the method **stub**)

Make sure you submit **MoreMath**. It'll be much like the `SimpleMathUtilities` class you saw in L04, except it'll have value-returning methods instead of void ones.

Details

This assignment is somewhat different than the ones that have come before. For the first time *I'm* providing *you* with the program. What *you* have to do is create and submit a class that implements some of the methods used in that program.

Do not modify the program I provide you with! If you do, there is a good chance that you will end up submitting code that does not work when we try to run it. If you do that, you will be *severely* penalized.

The class you need to create is called **MoreMath**. It mostly has some methods for doing mathematical functions (not provided by the `Math` class), but it also includes methods that return information about *you*. You need to create all those methods.

The class you are creating is **not a program** -- it does not have a `main` method. You should use NetBeans to create a **Java Class**, ~~not a Java Main Class~~.

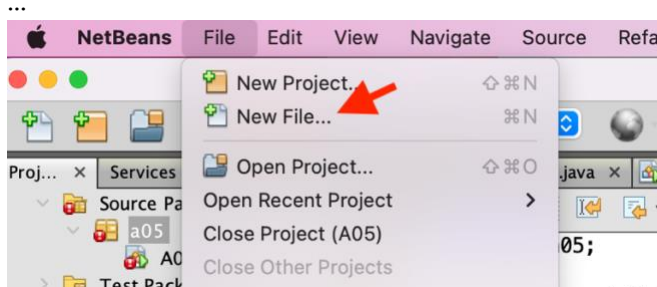
That means you can't run your class. If you try to run **MoreMath.java** you'll get an error message:

```
Class "MoreMath" does not have a main method.
```

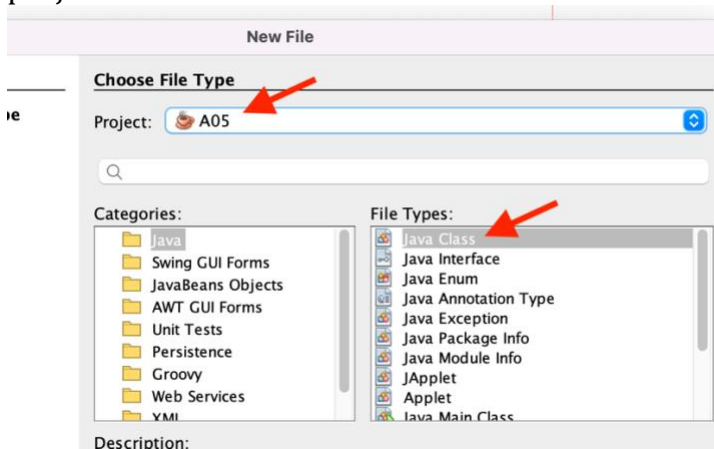
That's what's supposed to happen. If you *don't* get that error message, *then* you've got a problem!

What you want to do is:

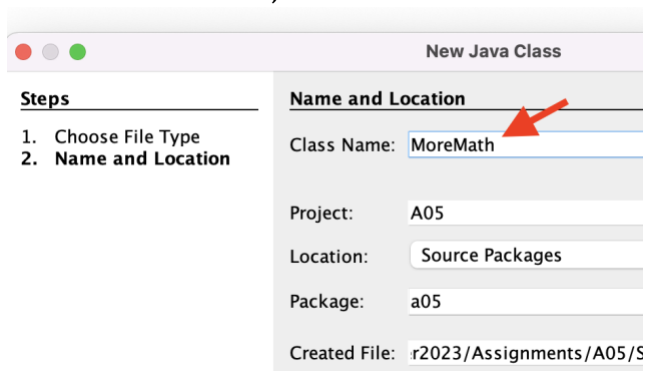
- 1- Create project A05 with a program named A05.
- 2- Replace the content of A05.java in your project with that of the file (A05.java) that I have provided for you.
- 3- While you are in A05.java file in NetBeans, from File menu, select New File



- 4- In the open window, choose Java Class and make sure A05 is the selected project.



- 5- In the next window, name the class MoreMath and press button Finish!



- 6- Make sure to add Javadoc at the beginning of this file with the class description and your name and A# after @author tag.

In general, you should be able to look at the commands in the program and figure out what methods you need to create. However, for this assignment, I will provide you with a list of the methods you need.

- **authorName()** -- returns the author's name.

For example, my version of this method returns "Somayeh kafaie".

- **authorNumber()** -- returns the author's A-number.

For example, my version of this method returns "A00000000".

- **log2(double x)** -- returns the log base 2 of x.

This is a simple calculation using the log function from Math. It's just the log of x divided by the log of 2.

- **randomInt(int n)** -- returns a random number in the range 1 to n.

This is another simple calculation that uses a Math function. For example, randomInt(6) returns a number between 1 and 6. (I've shown you how to do that before in lectures as well as L05)

- **factorial(int n)** -- returns the factorial of the non-negative number given to it.

This function requires you to use a loop. The factorial of n is $1 * 2 * 3 * 4 * \dots * (n-1) * n$, and is written $n!$. $0!$ is defined to be 1. For numbers less than 0, this function can return 1 (that'll be easiest).

- **fibonacciSum(int count)** -- returns the sum of the first `count` Fibonacci numbers.

The Fibonacci sequence follows the rule that **each number is equal to the sum of the preceding two numbers**. The Fibonacci sequence begins with the following 14 integers: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233 ... Each number, starting with the third, adheres to the prescribed formula.

This function requires you to use a loop and generate these numbers while you keep track of their sum. For example, when `count = 5`, this method will return 7 (which is equal to $0 + 1 + 1 + 2 + 3$).

For `count` less than 2, this function simply returns 0.

Notes

- Start by pasting the bold bits above into your **MoreMath** class.
- Use the non-bold bit right after the bold bit as your `javadoc` description on each method.

- Add the `@param` and `@return` tags to your javadoc comments. You *will* be expected to produce high-quality documentation in later CS courses.
- Figure out the most reasonable return type for each method and turn each of the bits you pasted into a *stub*. When you're done, there should be no red-squiggly lines in either file.
- Run the program **A05**.

When you press the green arrow and NetBeans asks you what's the "main class" of your project, choose **A05**. It *should* be the only option!

The output of your program will be (mostly) wrong, but it *should* run.

If the program does not compile and run -- *even if it says the problems are in A05* -- then the problem is *your code*. My code is exactly the way I want it to be, and you *mustn't change it*.

- Finally, revise each method to return the correct value. Run the program again after each method is completed to ensure that you have it done properly. (Compare your output to the sample output.)

In most cases the methods can be a single line, but **factorial** and **fibonacciSum** will require multiple lines and possibly local variables.

Grading Outline

- 60% -- Methods perform as required
- 20% -- Methods show good design
- 20% -- Submitted material meets the standard requirements.

Sample output

Please note that your output for `randomInt()` method will be most likely different from mine (it's supposed to generate random numbers after all!).

```
More Math Functions
-----
```

```
by Somayeh Kafaie (A000000000)
```

```
This program tests the methods in MoreMath
written by Somayeh Kafaie (A000000000)
```

```
CSCI 1226 -- Winter 2023
Assignment A05
```

```
MoreMath.log2(1.0) == 0.0
MoreMath.log2(4.0) == 2.0
MoreMath.log2(11.313708498984761) == 3.5
MoreMath.log2(1000.0) == 9.965784284662087
```

```
MoreMath.randomInt(6) == 3
MoreMath.randomInt(6) == 6
MoreMath.randomInt(10) == 1
MoreMath.randomInt(12) == 7
MoreMath.randomInt(20) == 19
MoreMath.randomInt(1000) == 264
```

```
MoreMath.factorial(1) == 1
MoreMath.factorial(5) == 120
MoreMath.factorial(10) == 3628800
MoreMath.factorial(-3) == 1
```

```
MoreMath.FibonacciSum(1) == 0
MoreMath.FibonacciSum(-1) == 0
MoreMath.FibonacciSum(5) == 7
MoreMath.FibonacciSum(2) == 1
MoreMath.FibonacciSum(10) == 88
MoreMath.FibonacciSum(20) == 10945
MoreMath.FibonacciSum(34) == 9227464
```