

Assignment 2

(Due date for assignment is Thursday November 14th, 4:00 p.m. ., Late date Monday November. 18th, 4:00 p.m.)

This assignment may be completed with a partner. Sorry, no groups of 3 or more.

Objective

The objective of this assignment is to investigate and use Junit testing. As an added bonus you will be required to use Github for versioning control. Thus it is important that you start early and organize your team to complete the assignment.

The Assignment

The java math libraries have all the trig metric functions. What if it didn't exist? You will be tasked to write your own trig library for java. In doing so you will need to implement all the standard functions, SIN, COS, TAN as well as their inverses. For completeness, you are to include COT, SEC, CSC. To calculate these functions consider Taylor series expansions. You might find the following resource helpful https://www.efunda.com/math/taylor_series/taylor_series.cfm. It should be noted that these series are based in radian measure. It would be nice that your library could also generate the function values from degrees. Thus your library will require the appropriate conversion as well to allow the user to chain together functions to produce the desired result. **You are not allowed to use any java math library functions. If you need a function, then you must create it.**

As part of your testing make sure that the functions are sound in all 4 quadrants. Will throw a custom exception for invalid usage. The class you will develop will be a static class.

Junit testing is a testing framework for Test First Development. This is an educational assignment, in that you will research and learn the Junit test suit and apply it to the class definition above. The below link is a tutorial for Junit. Its installation, and configuration to run an automated test on a code project. All of which you will be responsible for.

<https://www.guru99.com/junit-tutorial.html>

As part of the assignment you will be required to code the above class, integrating the Junit testing as you go. That is, you will also develop a test class under Junit to test the framework of your class as you implement the constructors and methods. When you are finished you should be able to run the test and verify that all cases have been covered, and that your class is sound.

What this assignment is not. Turning in a working class without any test suit will earn you a big fat zero. It is not about writing a java class, it is about learning Junit testing. It also means your class has to be correct. Thus, marks are assigned for completeness and thoroughness of the testing and the construction of the test environment.

Output should include the tests which are defined on non implemented methods and constructors. Thus initial tests should fail. You are to define this framework first, version this code as version 1, run it and save the version and output.

Implement the constructors and methods, show the tests at 1/3 complete, 2/3 complete and fully complete, along with the versions which match the tests. How about version 2.0, 3.0, and 4.0 as the final version.

Marking will be based on the quality of the submission which consists of the versions and test output as described above. That is, teams which submit a high quality assignment will be distinguished from the "night before it is due" submission.

Because of the size of this assignment you are given 3 weeks to complete it. Please start early.

GitHub:

As part of the assignment, you will produce a GitHub suit of versions, these should exist as ver 1.0 to ver 4.0 on the GitHub site. You will be given a tutorial on GitHub in a coming class. Marks will be assigned to this part as well.

Please add **bockusd** and **sajalcse** to your GitHub repository.

Submission

- Your submission should be contained in a large (8.5 inch x 11 inch) envelope.
- **Cover Sheet** completely filled out, available from: "<http://www.cosc.brocku.ca/forms/teamcover>" **Note:** your assignment will not be marked unless one is submitted with the assignment on the assignment due date. This should be stapled to the outside of the envelope.
- Print out of all code and all tests which have been run on your code.
- Clearly identify the code version and test output version.
- Clearly identify the GitHub repository for your code.

End