# CptS 233 Micro Assignment #2

## BST Adding and Removing Helpers, with some demo testing

For this micro assignment, you must fully implement the addElementHelper and removeElementHelper functions inside the BinarySearchTree class. All changes to the code must take place within "BinarySearchTree.java". The places where code needs to be added are noted with “MA TODO” comments.

If you build and run BSTAddRemove_main.java you’ll get a suite of tests on the BST implemented in BinarySearchTree.java. These tests exercise the interfaces, and if you successfully implement the MA TODO sections all tests should pass.

Just like in MA1, this assignment has been delivered to your Git repo. It is on a branch called MA2 BST Add Remove. You MUST go to the gitlab server and merge in the merge request created by this assignment. After that, you can run ‘git pull’ on your computer to get the new code to start working.

NOTE: If you use an IDE (Eclipse, etc) to edit your project, you’ll need to put the files back into the MA2-BSTAddRemoveHelpers directory once you’re done working. Various IDEs like to keep their files in various directory structures and I cannot build the testing system to detect and use them all. By putting the files back into the same places when you’re done working. it allows my GitLab testing code to work properly. If you don’t put the files back, it won’t build and test properly. Feel free to make more sub directories to work in as you see fit, but these files are in the final place they need to run from.

There’s a simple build script called “Makefile” in the MA2 directory. This is for a build system called ‘make’. If you’re on Linux or have XCode installed, you can run ‘make build’ and ‘make test’ to build and test the project. This isn’t required for the project to work, but it’s a shortcut. Windows people can use it too, but you’ll need to figure out how to install GNU make.

Your code must be added, committed, and pushed to your Git repository to be turned in. Put a small file onto blackboard to show the TA that you’re done working and need to be graded.

## Grading

Your submission will be graded based on the following:

1. [8] Your solution builds, does not cause any runtime issues, and passes all test cases
2. [2] Style points:
    - You provide meaningful variable names
    - You provide sufficient and has meaningful comments
    - Your code is well structured