CS1 - Function and Testing - Points Lab Possible Points: 110

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Built-in and User Defined Functions & Automated Testing Lab**

Write a C++ program that prompts the user to enter two points on a 2D geometry and finds the distance between the two points.

**Lab Instructions**

1. Open your repo folder (CS1-…) in Visual Studio Code.
2. Create a folder called **functions** in the labs folder.
   1. Inside **functions** folder, create two files: **main.cpp** and **Makefile**
3. Add new files to git repo; commit and push.
4. Do **add; commit and push** as often as possible after every major improvement or addition to your program so you are familiar with the commands and you’ve a working backup!
5. Use starter code stub main.cpp in **CPP-Fundamentals->labs->functions->points** lab folder as a hint to complete the lab.
6. Type, fix and use the Makefile in the lab folder to build and run your lab.
7. Never copy paste code; you’ll not learn anything by taking a shortcut!
8. Type some lines of code and use **incremental development** techniques to learn what the new code does; continue the process until you complete your lab
9. Fix all the FIXMEs and write #fixed# at the end of each code FIXME that’s fixed except at the end of your name and date.
10. Lab has a bonus FIXME. **(10 points)**
11. Run and manually test your programs many times.
12. Create a screenshot showing the complete run of your program and add it to the repository. (**10 points**)
13. Update your README file **(10 points)** as shown here: <https://github.com/rambasnet/csci000-astudent>
14. All FIXMEs are worth equal points unless stated otherwise.

=========================================================================

**Submission:**

1. Add all the relevant source file(s) and documents into the correct folder and do a final add, commit, and push before the due date.
   1. $ git pull
   2. $ git status
   3. $ git add <filename>… - add each file that was new or modified that is part of this assignment
   4. $ git commit -m “Final Submission”
   5. $ git push
   6. $ git status
2. Check and make sure the files are pushed to your GitHub repo.
3. NOTE: Do not add and commit to this lab folder after the due date as it may be considered late submission!