Programming Rules/Prof. Stamos

Read this so that you will not be surprised later.

I am willing to point you to the right direction by answering any questions regarding your assignments, do not hesitate to send me an email. It is **extremely important** to follow the following rules. Failure to follow them will result into a lower grade for your assignment, and in many cases could result to zero credit. Read the following carefully.

Please follow the following rules:

1. All assignment submissions will happen through blackboard. You will be submitting the source code and other documents as a single archive (.zip or .tar.gz) as described below. For information of how to submit assignments through blackboard visit: https://help.blackboard.com/Learn/Student/Assignments/Submit_Assignments

2. To submit an assignment you must submit electronically via blackboard a zipped (.zip or .tar.gz) directory that contains the following (do not submit .rar archives or .7z files):
    a. the program source code (all source and header files)
        it is important that each source file starts with a commented header that includes your name
    b. your Makefile. The use of a Makefile is **mandatory**. All programs should compile by typing "make all". Failure to provide a Makefile will result in a penalty in your assignment grade.
    c. A README.txt file in which the following are explained:
        i. Which parts of your assignment were completed
        ii. Any bugs that you have encountered
        iii. Complete instructions of how to run your program(s).
        iv. the input file (if any) and the output files (if any).

    For sample Makefile and sample source code please see the Sample Code under blackboard. Please download the sample files and understand how a Makefile works.

IMPORTANT: *Please make sure that the name of the submitted directory is your full name. For instance name the directory as John_Adams_AssignmentX.*

IMPORTANT: *Do not create any subdirectories under you top directory. All your source files should be in ONE directory.*

3. Your program should **compile and run** correctly in one of the Linux machines of 1001B Lab at Hunter North. See the end of this message for information about the lab. All lab machines have the same configuration. You should have an account in the 1001B Lab. If you do not have an account please contact me before the second class of the semester. Note, that you can access the lab machines remotely. Also note that **we will not debug** your program. **We are not going to alter your code** in any way. Your code should compile and run as submitted. The **OFFICIAL compiler** for this class is g++ and the version installed in the lab machines (output of g++ --version):

is

g++ (Ubuntu 5.4.0-6ubuntu1~16.04.11) 5.4.0 20160609

4. For full credit, an assignment must be submitted no later than midnight on the day it is due.
   For each late day it loses 10 points. After four late days the assignment **will not be accepted**.
5. The program must be **your work and your work alone**. Attributing someone else's work as your own is plagiarism, which is a violation of College policy and an unethical activity. Contract cheating is a form of academic dishonesty in which students get others to complete their coursework for them. Please read more information on Contract cheating from http://en.wikipedia.org/wiki/Contract_cheating
   Please note that you can use the source code associated with the textbook unless otherwise mentioned in the assignment. You can exchange ideas and suggestions with your fellow students, but you are not allowed to copy code.

The grade of your assignment will be based on the following:

1. Every program must be correct to receive full credit. "Correct" means that for every possible input, it produces output that is consistent with the specification. If the program produces correct results for some, but not all, inputs, it is not correct. Correctness is usually 50% to 60% of the grade.

2. Every program must satisfy specified performance requirements. This means that it uses an amount of storage and running time within specified or reasonable limits.
3. Every program must be well-designed. This is a subjective criterion. There is no one way to design a program well. There are, however, commonly accepted standards of what "well designed" means. These include:
   a. using appropriate ADTs and algorithms,
   b. decomposing the program into appropriately-sized modules that are cohesive,
   c. that decisions about which objects is exposed and which are hidden are justified by sound arguments,
   d. that classes are trim but adequate in terms of the methods they provide,
   e. that there are **no** global or static variables unless the problem cannot be solved without them, and
   f. that the design corresponds to the problem statement in a way that can be explained in the documentation.
   There are no rigid rules. All rules can be broken, but it is best to check with me before taking a big chance. A program's design is worth from 10% to 25% of the grade.
4. Every program must be professionally documented. Every distinct source code file must contain a preamble with the file's title, author, brief purpose, date of creation. All functions must have a prologue containing comments for each parameter and appropriate pre-and post- conditions in the header file (do not provide comments on top of functions in the implementation file). All non-trivial algorithms must be documented in plain

English in a multi-line comment block. All non-trivial declarations must have adjoining, brief comments. There is no need to add documentation for obvious declarations. Documentation is worth 10% of your grade.

5. A good resource for C++ style is the one provided by Google: https://google.github.io/styleguide/cppguide.html

1001B Computer Science laboratory

The 1001B laboratory is located on the 10th floor of Hunter North. You should have access using your Hunter One Card. The laboratory consists of 29 Linux machines (running Ubuntu). You should all have accounts in the laboratory. If you don't have an account please contact me directly. Your assignment should compile and run in one of the lab machines. Of course you can work on your own machine, but before submission you should make sure that your program(s) compile and run in one of the lab machines. If you have your own Linux environment please note that the compiler we are going to use is g++ and its version is

g++ (Ubuntu 5.4.0-6ubuntu1~16.04.11) 5.4.0 201606

To check the version of your compiler please type g++ --version

You can also remotely login to the lab machines as follows:

1) ssh <your_username>@eniac.cs.hunter.cuny.edu
2) type your password
3) Now you are at a gateway machine that is called eniac
4) Do not do any processing on eniac. Just ssh through eniac to one of the machines in the lab (see next step)
5) ssh <your_username>@cslab<X>.cs.hunter.cuny.edu, where <X> is the number 1 through 29. You can pick any machine. If the machine is down you can try another machine. For instance to login to the 2nd machine type:

   ssh <your_username>@cslab2.cs.hunter.cuny.edu
6) All cslab<X> machines and eniac see the same directories for your account. That means that you see the same files in all machines.
7) If you want to test your programs in one of the cslab machines you can use sftp in order to transfer your code to eniac. Then you can ssh to eniac (see step 1) and after that ssh to any cslab<X> machine (see step 5).