CS 433 Programming Grading Policies and Submission Guidelines

Grading

Your programs should be written in C/C++ unless otherwise specified. The programs must be able to compile and run on the class server "cs433.cs.csusm.edu". Your programming assignments will be graded according to following criteria:

1. Correctness (60%): If your program works as required, you get the full points. You may get partial credits.

2. Design (20%): How well your programs are structured and implemented according to common programming guidelines, including but not limited to

   - Check for validity of input data to program and functions to prevent abnormal behaviors, e.g. program shouldn't just crash if input values are invalid, but should exit graciously and print clear error information.
   - Programs should not be built around a specific dataset but should work correctly for any reasonable test dataset.
   - Programs that use threads should ensure that their functions are re-entrant.
   - Programs should be robust, for example to avoid comparing equality of floating point numbers.
   - Function coherence: each function should correspond to one sub-task in the overall algorithm to solve the problem.
   - Function independence: each function should be independent, i.e., it should be self-contained and it should perform its task successfully without needing to know the inner workings of the calling function.
   - Use appropriate parameters rather than global variables (stay away from global variables whenever possible). Avoid side effects.

3. Documentation and Report (20%): Provide sufficient comments so that others can understand what you have done. You should also describe in the report how to compile and run the programs if it is not obvious. Follow the guidelines below to comment your code:

   - Program should begin with a preface stating:

     1. Programmer's name.
     2. Date written.

3. Course and assignment number
4. A summary or description of functionality, i.e., what the program does and a brief description of how it does it.
5. Any known assumptions built into the program.
6. Any special instructions on how to run it.

- Each function should have a brief comment explaining the purpose of the function, input parameters to the function, and output. The comments might also state the limitations of the function using possibly pre-conditions and post-conditions.
- Most variables should have a brief comment explaining their purpose. Looping control variables such as: "`x`" or "`i`" need not be commented.
- Comment your code to explain tricky and important points (not every line).
- Comments should be comprehensible and useful. Comments such as: "`This adds 1 to Count`" are trivial and should be avoided.
- Comments should be clearly organized and easily distinguishable from the actual code.

You should also write a report for each programming assignment, which should clearly mention your names, a descriptive list of the submitted files (e.g. which are the executable and source files etc.) and the instructions to compile and run your program. If applicable, your report should include any results for the assignment, the features implemented or missing from the submission, and describe the design decisions and implementation choices you made for the project and lessons learned from this assignment. You should give references to the code you used from online/book resources and explain your contributions. You may also discuss extra things you have done in the project, future improvements to your program, and any comments about the project that you want to communicate to the instructor. Your reports are usually expected to be 2 to 4 pages and contain sufficient details.

Submission Guidelines

For every programming assignment, you should submit at least the following files:

- Source code
- Executable
- Makefile
- Any data files used in the program
- Report

Your programs should use a **Makefile** so that I can compile it by using the `make` command. An example Makefile is provided with assignment 1. Your program MUST compile in order to get points for the project.

You should first edit and debug your program in your home directory on the server cs433.cs.csusm.edu. When you are ready to turn in your program, please follow the steps below:

1. Change (`cd`) to the assignment submission directory, which is `/cs433/assign1` for assignment 1 or `/cs433/assign2` for assignment 2 ..., and make a directory (`mkdir`) for your group. If you work alone you should use your login name as the directory name, e.g. **"`smith001`"**; if you work as a pair, use both of your login names as the directory name, e.g. **"`smith001_n_doe001`"**. Use command **"`chmod 700 dir`"**, where **"`dir`"** is your directory name, to change the permission of your directory so that other students cannot read your files. *It is critical to make a directory and submit your programming assignment in the submission folder, NOT your own home directory!* We will not be able to access your home directory.

2. Copy the source code, executable, Makefile and any other required files into the directory created in step 1. You should double check that your program can compile and run in the destination directory. You can replace the files in the destination directory if necessary, but don not touch it after the due date.

3. Submit your report as a PDF file on cougar courses (cc.csusm.edu). Each group only needs to submit one copy.