Project preface

PREFACE

- **Project teams:** This project is designed to be worked on by a team of programmers. **Up to 3 students (up to 4 in summer 2020)** may organize themselves into a project team.

  If all else fails, let me know and I will assign you to a team. Any team of two students has a good chance of me adding another student to your team, whether you like it or not. **Your team should be made by the third week, and submit team information here (https://canvas.uw.edu/courses/1408528/assignments/5665687) .**

  It is not uncommon for project teams to basically fall apart and self-destruct. Here is a hand-out (https://canvas.uw.edu/courses/1408528/files/67695508/download?wrap=1) from Professor Walter Freytag (Business School) that I encourage you to read and discuss among yourselves at the start of your project. It won't guarantee a successful project, but reading it can help you set reasonable expectations for you as a team and as individuals within that team.

  1) You can voluntarily leave your team. In that case, you can have a new team.

  2) A group can fire a team member. In that case, the fired person has to work alone, without creating a new team.

  After two weeks, you cannot change your team, and you will be a team no matter what.

  Warning: Project can be graded individually if necessary. If any member has not contributed to the programming, and the concern is expressed in the evaluation report, then the individual will get deducted points. Please note that, your final project is graded based on your final product. Then, from the individual report, the person who significantly harmed the team's work will get point reduction.

- **Progress reports:** Working for a project in a team is not easy. There are many issues, but most importantly, you should manage your time well. Every two weeks, you should submit a team progress report as a team. For the progress report guide, please click **here. (https://canvas.uw.edu/courses/1408528/pages/progress-report)**

- **Scope of the project:** This project is a substantial effort, but it really isn't the "ordeal" that some students turn it into. For each of you, it is about the effort required to do 2-3 homework problem sets. Please approach this systematically, and do not wait too long to get started.

- **Cheating:** This project has given before. You may have friends who have taken this class before you and may still have a copy of their project. Copying the project, or part of the project, from a former student is cheating and will not be tolerated. **Don't even think about it**! There are a special program developed that scans every past project into a database. Your project will be

compared to this database of every former project. Believe me, the program works. I can even tell if you just copied a few lines of code from someone else, who you copied from and what you copied. You can't fool the program by changing the comments or the variable names. Also, I have the right to change the grade of the person you copied from. Don't do it. It isn't worth it.

There is one exception to the above rule. If you are repeating this class, you may make use of **the particular part of the project code that you previously developed**. You may not reuse the project code developed by your former teammates. You may only use code that you wrote. You may not share the entire project contents with your new project team. That would be considered to be cheating on their part and cheating on your part for providing it. Note that I will detect the fact that you have reused code with my cheating program, but I will see that it was your previous work when I review it. However, if you did not write any code with your previous project team, you are not entitled to use any of the code written by your former team mates. Remember, I have your personal statements on file and I can check to see what code you wrote the last time around.

In summary, if, for any reason, you are repeating the class, you may only re-use the code that you personally wrote.

- **Testing:** I will use the Easy68K simulator to load your program into memory at $1000. I will then load my test program somewhere else in memory. My program will be comprised of a set of the instructions and data that are given in this specification. The test program will include other instructions that you are not required to decode, data fields, blank space, etc. All of your output will be sent to the display and also logged to a file. I then compare your output with my listfile.

  When I grade your program I check it in two ways. First, I run the program and use it as any user would. I test the user interface and check it for robust recovery from input errors. Next, I ask it to disassemble my test program located somewhere in memory. For this phase of the testing I log all the screen output to a file. You can do this as well, and you should.

  At this point all I/O between the keyboard and the screen will also be sent to this file. You can then review your own file and see all of your I/O. It saves lots of time.