

Capstone Team Project for summer 2016 (*online*) – Learning Objectives and How You Will Be Graded

Learning objective #1: Learn the *process* of developing code in a team project.

You will succeed in this learning objective by *doing* a process that is similar to what real software development projects use. 40% of your grade on the project is based on this learning objective. I will assess it by seeing whether you:

- Work in 3 Sprints. At the beginning of each Sprint, you list the Features that you hope to accomplish in that Sprint, with guesses for the subsequent Sprints. At the end of each Sprint, you evaluate to what extent you implemented the planned Features, and re-evaluate and make more explicit the Features you plan for the next Sprint.

You write your Sprint Plans using a tool called Trello. Each Sprint Plan will be very short and simple, but required.

- Include short but meaningful messages with every COMMIT that you do.
- Meet regularly (but briefly) with your teammates.
- Work using iterative enhancement. That is, you repeatedly:
 - Take a small part of the feature that you are implementing, get that to work, and then commit that work.

I expect *frequent commits*, but always of *working* code.

- Record all your hours-worked in your *hours.txt* file.

Learning objective #2: Learn how to work with teammates on a software development project.

You will succeed in this learning objective by *doing* it. 10% of your grade on the project is based on this learning objective. I will assess it by seeing whether your code successfully integrates with the team's code. Additionally, each team member will do a short survey that includes questions about how effective a teammate you were.

Learning objective #3: Deepen your understanding of the concepts we have studied in this course.

35% of your grade on the project is based on this learning objective. I will assess it by seeing:

- Did you implement your Features correctly?
- Is your code high-quality: Does it use function calls and/or loops to avoid repeating code unnecessarily? Are function and variable names meaningful? Do you use blank lines to separate chunks of your code, but without lots of extra blank lines? Does each function that you write have a short comment that states what that function does (WHAT it does, NOT HOW it does it)? Did you include your name in your module, in the comment at the top?
- Did you use functions effectively? Loops where appropriate? Lists where appropriate?
- Did you read **data** from files instead of putting **data** into your code? (For example, if you write code to play a song, the notes and their durations should be in a file, and you write code that reads the file and makes sounds based on the data.)

Learning objective #4: Learn how to learn ***new applications*** of the concepts that we have studied.

15% of your grade on the project is based on this learning objective. I will assess it by seeing whether:

- Did you make a Graphical User Interface (GUI) that is event-driven and uses something more than just buttons and entry boxes?
- Did you implement at least one Feature that required you to learn new things from the Internet and/or examples that I supplied?
- To what extent were the Features that you implemented challenging?
- Did you work an appropriate number of hours on the project, as entered in your *hours.txt* file?