Course Project

Info 206

Objectives

Throughout the duration of the course teams are expected to work together towards completion of a software development project. Teams are responsible for:

- identifying the problem that they aim to address
- working collaboratively and sharing code through Github
- building and testing software
- presenting project updates twice to the class
- display final project results on the last day of the course (17 October)

Expectations

Through your team project, you are expected to write original code in Python to build a computer program that aims to address a problem identified by the team. In doing so, teams may utilize existing modules from Python's standard library or

Remember that you have limited time to work on the project - final presentations are due on October 17, 2017 - thus teams should identify a problem and coding tasks that are reasonably scoped to be completed within the time frame of the course schedule. You are welcome to discuss your project ideas with the instructor in order to assess whether or not your project is feasible within the allotted time frame.

- The project will require contributions (observed through Github commits) to a team project. Individual team members will be assessed for their level of participation through their recorded activity on Github. (The instructor also reserves the right to solicit information from the team to assess the level of involvement of individual team members.)
- This project must not exceed 2000 lines of original code.
- You are not likely to need more than 10 classes
- While teams have the option of using existing code from other sources (e.g. forking a repository, using user written libraries), teams must explicitly obtain instructor permission if using anything outside of the standard Python library.
- Please include all references that you used to build out your application.
- Included with the code you should do a ~one page write up of what you completed, any you challenges you faced, as well as how we should go about testing and using your project code. This information can be included in the README file of the Github repository or as a separate document contained within the repository.

Deliverables

In preparing projects, teams will be expected to complete the following deliverables:

- Submit a problem statement in which the team identifies the issue that it seeks to address through a computer program.

  In this document, you can write an overview of the computer program you want to write. The statement should be 1-2 pages.

  A final version of your problem statement is due via your team GitHub project folder by due September 12, 2017.

- Submit a codebase outline (or design document) in which the team identifies the issue that it seeks to address through a computer program.

  In this document, you are expected to present the structure of the modular code that the team will write. You are not expected to have written full code at this point although you may do some degree of pseudo-code to help explain how you intend to formulate the code.

  A final version of your design document is due via your team GitHub project folder by due October 05, 2017.

- Brief project updates to the class during the course of the semester. (Time permitting)

- Submit a fully-functioning codebase in which the team implements a solution to the targeted problem.

  Teams are expected to have a set of modules and classes that are readable, replicable, and written according to Python programming best practices.

  A final version of your codebase is due via your team GitHub project folder by due October 17, 2017.

- On the final day of the course, we will have a lightning presentation round in which teams have 5 minutes to present their project to the entire class. Due to the short period available, teams are encouraged to focus on:

  1. summarizing the problem statement succinctly;
  2. running the computer program;
  3. summarizing any challenges and learnings that the team had in writing and/or implementing the program.

  All other teams will be able to view your team's codebase, thus you are encouraged to refer classmates to your Github repository to view how the team tackled the programming.

Team Formation

Teams will consist of approximately 4 individuals. You will be able to select one team member to work with on the course-long group project. Each pairing will be randomly assigned with one other pairing to form the complete project team. Teams will be announced by September 5.

Important Dates

- Project Status Update #1 - Problem statement (12 September)
- Project Status Update #2 - Codebase outline (O5 October)
- Final Project Presentations (17 October)