Programming for Biologists and Biochemists

Goals:
The goal of this project is to provide you with an opportunity to apply your knowledge to solve an open-ended research problem in biology. The task is to design and build a Python program that can be used to investigate a biological research question.

Purpose:
The purpose of this project is to give you experience developing a tool that can be used to help answer a specific research question. You will gain this experience by designing a Python program that controls the flow of information from obtaining input data (from file(s), the web, the command line, etc.), manipulating this input, and writing out results (to the screen or file(s)). While the Python programming language will be used to control this information flow, you are not limited to Python functions or modules (standard or ones you write) to manipulate the data (e.g. Python can call other programs to operate on the data).

Collaboration:
This is an individual project. Each student is expected to work with a faculty advisor or other mentor\[1\] to help define the specific research question their program will address. During the course of outlining the project scope, students may solicit help through collaboration with faculty, other students, the TA or the instructor. This project is designed to be a learning experience so you are encouraged to work with other students or borrow existing code whenever possible. However, any collaboration or borrowing of code should be acknowledged with proper credit given to the source and other people’s effort in helping you with the project. That said, the majority of the code design and implementation should be an individual effort.

Deliverables:
The grade for this project is based on three deliverables.

1. 3 minute (maximum) oral presentation of your program being used to solve the research question you are addressing. To receive full credit on the presentation portion, you must:
   
   (a) Provide quick overview of problem that you are addressing. This overview should be brief and use language so that everyone in the class will understand your research question.
   
   (b) Provide a flow-chart or diagram of how your program works.
   
   (c) Show us one result from your program and the main conclusion or take-home point.
   
   (d) Have no more than three slides!

2. 5–10 page report or writeup of your project. To receive full credit on the report, you must
   
   (a) Introduce the research problem you are addressing, why this question is important and how programming will help answer this problem.
   
   (b) Describe the programming concepts and algorithms that were used to complete the program.
   
   (c) Outline the major results of your code
   
   (d) Discuss and interpret
     
     i. How well the code performs the specific tasks you implemented.
     
     ii. How the program could be extended to (a) address problems related to your specific question and (b) answer unrelated problems that use the underlying programming concepts/algorithms that you implemented.

3. Python program that addresses your biological research question and any necessary input files (modules and/or data) used to run the program.

Checkpoints:
Checkpoints are due to your dropbox the instructor and the TA via email unless otherwise indicated.

1. Problem statement and overview:
   
   Friday April 29, 2011 by 11p.m.

\[1\] If you are not currently working with a faculty advisor or outside mentor, please see the instructor and the TA so we can assign you a mentor.
2. First Check Point:
Friday May 13, 2011 by 11p.m.
- Each person should turn in 2–3 pages (maximum) that include (a) a flow-chart of your program, (b) the pseudo code of your project, and what you are stuck on at that point in time.
- Create a folder called check_point1 in your Drop Box project folder to turn in your assignment there.
- During class-time on May 17, we will conduct a design review. The instructor will put everything onto a single computer that will be used to project the information on screen in class.
- You will have 2 minutes (maximum) to walk us through your ideas. The other members of the class and the teaching staff will be on hand to hear about your ideas and provide feedback and advice.
- Accepted formats for submission include .doc(x), .pdf, .key or .ppt(x) only!

3. Second Check Point:
Friday May 27, 2011 by 11p.m.
- Each person should turn in their code that demonstrates a complete main program that has:
  - the ability to read in whatever input data you are using for your project
  - proper flow of control through the program
  - two working functions that perform steps along the way to the final result(s).
- The Python code, input data, and instructions for running your code, along with all expected results from your functions should be submitted into your Drop Box, in a folder called check_point2 in your project folder on eCommons.
- Accepted formats for submission include .py, .txt, .doc(x) or .pdf only!

4. Public Presentation:
Tuesday May 31, 2011 by 2p.m.
- Presentations will take place during class-time on May 31. Each person will use one of the provided slide templates (key or ppt) for their presentation. Slides should be put into your Drop Box, in a folder called presentation in your project folder on eCommons ahead of time and will be put onto a single computer for class.
- You will have 3 minutes (maximum) to give your presentation. If you finish before the 3 minute time limit, we will allow questions from the audience. We will have a playful mechanism for cutting you off if you exceed 3 minutes.
- Accepted formats for submitting slides include .key or .ppt(x) only!

5. Final Code:
Friday June 3, 2011 by 11p.m.
- The final version of your Python program and any supporting documents to run the program should be submitted into your Drop Box, in a folder called final_code in your project folder on eCommons.

6. Report:
Draft due on Monday May 30, 2011 by 11p.m (submit to your Drop Box, in a folder called draft in your project folder on eCommons). Final version with revisions due by 11p.m. on Monday June 6, 2011 (submit to your Drop Box, in a folder called final in your project folder on eCommons).
- Accepted formats for the report include .doc(x) or .pdf only!
Grading:

☐ Check-Point Performance (25%) based on demonstrating the required functionality at the checkpoints.

☐ Public Presentation (25%) based on the delivery of the presentation.

☐ Performance (25%) based on the results of the final code submitted.

☐ Report (25%) based on an evaluation of the written report. It will be judged on clarity of explanations, completeness and appropriateness of the documentation.