

Advanced Programming

Homework Assignment 8

Due Wednesday, March 19, at 3PM (first round)
and Wednesday, March 26, at 3PM (second round)

Preminimaries

1. Find a partner that you have not worked with yet. As with previous team assignments, all the programming for this one is to be done “shoulder to shoulder” with your partner.
2. Choose any programming task that interests you, subject to the following constraints:
 - Be a little bit unambitious in terms of the functionality you aim for, so that you have time to focus on making the code beautiful. Plan to spend maybe 10-15 hours on actual coding.
 - Choose a project that can be sensibly decomposed into several modules, and where it makes sense to construct significant tests for at least most of the modules.

If you are having trouble coming up with a good idea, drop me an email and I can suggest some possibilities.

Assignment

This assignment will be handed in *twice*.

1. In the *first round*, you will submit just a “sketch” of your program, where the overall task has been broken into modules, the main functions in each module have been specified (by writing down their type signatures) and listed in the module’s interface, and testing code for the main functions has been written, but none of the functions are actually implemented.
 - Your code should compile and run (i.e., it should provide both `main` and `test` actions, as described below), but it does not need to do anything interesting. In particular, the test actions will simply fail at this stage.
2. In the *second round*, you will fill in your sketch and submit a final, working program.
 - In the process of filling in functionality, you may find that your modular decomposition is unsatisfactory. It is fine to change it.
 - Similarly, you may find at this stage that some of your test cases were wrong, or that you need to add new ones. This is also fine.

Grading

1. This assignment will be graded primarily on the basis of clean design and coding. Cool functionality is a secondary concern.

2. Your primary aim should be to make your code as *readable* as possible.
 - Choose a clean decomposition into modules.
 - Make the coupling between modules as loose as possible.
 - Organize each module into “interface” and “implementation” parts, as discussed in class.
 - Include enough comments to help the reader see the overall picture of what’s going on, but don’t over-comment. Instead, aim to make the code self-documenting wherever possible.
 - Include testing code that illustrates the functionality of each major function in each major module. (It’s OK to omit testing code for a few modules where the tests would be very awkward to specify, like GUI code.) Each module should export an action `test` that executes all of its tests.

Submission instructions

- For each round, group your code into a single directory of `.hs` files. Name this directory `YourNames8`.
- Include a file `README` that gives a brief overview of the structure and tells the reader in what order they should read the rest of the files.

Put your names at the top of this file. Also, please put the approximate number of hours that you spent on this assignment.
- Your submission should define a module `Main` that includes an action `main` that makes your program perform its main function and an action `test` that runs *all* test functions in all modules. (This means that we can do

```
echo Main.test | ghci Main.hs
```

to run your testing code.)

- Email the whole directory to both `jschorr@seas.upenn.edu` and `bcpierce@cis.upenn.edu`.