CSCI 3060U - Winter 2025

Phase #4 - Back End Rapid Prototype

Due Monday, March 24, 2025 (11:59pm)

Phase #4

In this phase, you will design and rapidly program the first version of your new team's Back End. Since the next assignment will involve testing it, do not do full testing of your Back End at this time (no marks are for correctness yet!)

Before starting Phase #4 you will need to form a new group of 3-4 students from your lab section. Your new group cannot have any of the same group members as your previous group (Phases #1-3).

For the Back End you will continue to develop the Banking System in Python. All development for the Back End will need to be done using the PyCharm IDE (https://www.jetbrains.com/pycharm/). This is necessary to support the unit testing in Phase #5.

What to Hand In

In this phase, you will hand in:

1. A design document for your Back End, giving the overall structure of your solution, showing the classes and methods in a UML class diagram and a brief (one sentence) description of the intention of each method and class in a table.
2. The first version of the source code to implement your design. This version should run on at least some inputs but should not yet be completely tested (since that will be the next assignment, and it's better to leave some failures until then).

Your solution to this phase will be judged on the clarity and readability of the design and the code, so work at using your best programming practices, including naming variables, classes and methods meaningfully and commenting liberally to make it easy for another programmer to understand.

Your solution to this phase will not be judged on its correctness. This is to be a rapid first version, not a final product. Therefore, design and program it to work correctly, but don't worry about getting it fully debugged or tested yet since that's what we'll do next.

Marking Criteria

Phase #4 will be marked according to the following criteria.

1. Design (Primarily in the design document)

Architecture — 2 marks
- clearly documents structure of solution
- explicitly describes intention of each class and method
- accurately reflects solution structure
- clearly shows where inputs and outputs fit in

Completeness — 2 marks
- evidently addresses all required functionality
  (solution has parts to address all required operations and results)
- has specified inputs, outputs and files (only!)

2. Source Code (Details of the source code itself)

Structure and format — 2 marks

- code is structured and formatted such that architecture is clearly visible in code
- naming of classes and methods clearly reflect their role in the solution
- as little cloning or redundancy as possible

Maintainability — 2 marks

- avoidance of coding tricks and hacks
- simplest solution possible, no frills and extras
- clearest solution possible, no gratuitous optimizations

Internal documentation — 2 marks

- clear naming of all variables and constants to reflect their role in the solution
- comments at beginning of every class and method clearly documenting their interface and intention
- comment at beginning of main program documenting overall program intention, input and output files, and how the program is intended to be run

TOTAL **10 MARKS**