Assignment #3

Eric Rouse; Individual Programming 53

Understanding the Problem

Write a program that calculates the grade of a student based on the following parameters:

- Labs – 10%
- Quizzes – 10%
- Assignments – 40%
- Tests – 30%
- Final Project – 10%

And the user gets to pick how many of each thing there was. The user must be prompted for a score for the amount of times they answered for each thing.

Devising a Plan/Design

Here is the pseudocode that I came up with.

1. Prompt the user:

   Here are your options for class evaluation categories:

   q – quizzes

   a – assignments

   l – labs

   t – tests/exams

   f – final project

   d – done entering categories

2. Prompt to select a category. Keep prompting until all are selected (allow to skip final project if they want). For example:
   a. **Please select a category [q, a, l, t, f, d]: a**

      Please enter the number of assignments: 8
   b. **Please select a category [q,a, l, t, f, d]: t**

      Please enter the number of tests/exams: 2

3. Prompt user for specific scores in each category.
   a. Assignment 1:
   b. Assignment 2:
   c. …
   d. Quiz 1:
   e. Quiz 2

4. Calculate score based on the weights:
   - Labs – 10%
   - Quizzes – 10%
   - Assignments – 40%
   - Tests – 30%
   - Final Project – 10%
5. Print the Class average.

Error Handling:

- Print an error message if the user doesn't select all the categories, **excluding the f option, which is optional**.
- Print an error message if the user doesn't supply an integer for any one of the category values, **this excludes the f option, which doesn't have a number value**.
- Print an error message if the user enters a zero for any of the categories.
- Print an error message if the user selects an invalid category option.
- Print an error message if the user doesn't supply a number for the scores for the items.

Looking Back/Self-Reflection

I still don't really understand static and dynamic typing or when the "this." operation is necessary. Obviously I need to review some lectures. Also, I don't know what a good/bad amount of encapsulation is. It would be nice to get some feedback on this assignment to let me know some good demarcation lines.

I like how this method of programming allows all the user interaction/error handling to be streamlined, that was the worst part of the assignment last time.

It also made testing easier as it required testing each piece only once. And once you had a method working you could copy it and know it would work the same way. Reusing methods from the labs and other assignments, etc.

To check the output I worked through every possible path in the code and entered faulty data where I could.