CSCA08 Exercise 2

Due: Sept 30, 2016. 5:00pm

This week, we're working with functions and the design recipe. Your code will be auto-marked, so we won't have any way of knowing whether you actually followed the recipe, but a) if you decide not to follow it, you're only harming yourself, b) your TAs will not help you with step n of the recipe if you haven't completed step $n-1$, and c) we reserve the right to randomly check your commenting and design on any exercise with no prior warning.

For this exercise, you must add 4 functions to the file ex2.py[1]. (As always, all file and function names must be **exactly** as described in this handout). I've left some global variables for you to use in your calculations at the top of the file, feel free to play around with them, (you should be using these in your calculations), but return them to their original values before submitting.

1 Percentage

Complete a function called `percentage`, which takes as its parameters two floats [2], and returns the a number representing the first parameter as a percentage of the second parameter. We've provided you with the first few steps of the Design Recipe to get you started.

2 Term Mark

Write a function called `term_work_mark`, which calculates your term mark (as a contribution to your final grade) given your marks on all of the coursework components). Using the default weights this will return a maximum result of 55 (since the exam is worth 45% of your final grade):

- your mark on assignment 0 (by default this is out of 25)
- your mark on assignment 1 (by default this is out of 50)
- your mark on assignment 2 (by default this is out of 100)
- your mark on the exercises (by default this is out of 10)
- your mark on the quizzes (by default this it out of 5)
- your term test marks (by default this is out of 50)[3]

In the starter code provided in `ex2.py`, there are already several global variables that you will need to compute this score. The contribution of each piece of coursework is computed as:

$$\frac{mark-you-received}{maximum-mark} * weight$$

Your coursework mark is the sum of these component contributions. An example output is given below:

```
>>> term_work_mark(25, 50, 100, 10, 5, 50)
55.0
>>> term_work_mark(20, 45, 70, 8, 4, 40)
43.9
```

[1] there are already two functions and some global variables in the starter code, you do not need to edit this part of the file in any way

[2] remember, an int is just a special type of a float

[3] I'm not guaranteeing that these numbers will be correct for the real course, that's why I've left the variables there for you to change.

Hint:

Rather than writing the same computation over and over, you can make your life easier by using a helper function... perhaps some nice person may have left a useful function in the file before uploading it...

3 Final Mark

Write a function called `final_mark`, which performs a similar calculation to `term_work_mark`, but takes an additional parameter, your final exam mark (be default, out of 100), and returns your final mark for the term out of 100.

An example output is given below:

```
>>> final_mark(25, 50, 100, 10, 5, 50,100)
100.0
>>> final_mark(20, 45, 70, 8, 4, 40, 73)
76.75
```

Hint:

Be lazy! If you use the other functions you've written, this shouldn't take more than 3-4 lines of code.

4 See If You Pass

Write a function called `is_pass` that takes the same parameters as `final_mark`, and returns a boolean representing whether you passed the course. Remember that in order to pass the course you must get a final overall mark of 50 or greater, and a final exam mark of 40 or greater.

An example output is given below:

```
>>> is_pass(20, 45, 70, 8, 4, 40, 41)
True
>>> is_pass(20, 45, 70, 8, 4, 40, 39)
False
>>> is_pass(10, 21, 12, 2, 1, 15, 23)
False
```

Hint:

See hint for the previous section. 3-4 lines of code should be plenty. Don't do more work than you need to.

5 Testing it All Out

Now for the fun bit. On the course website is a file called `ex2_gui.py`. Put this in the same directory as your `ex2.py` file and run it. If you've done everything correctly, it should provide a GUI[4] for you to test your functions. You don't need to edit the code in the file or even read it (I wouldn't expect you to understand most of it yet anyway). But for those of you who are desperate to build something with buttons... click away!

[4] Graphical User Interface