CS50—Assignment 2
Functions

Due Monday, February 2nd at 11:59pm

hey nathan how is that project going?
just finished it about 20 minutes ago
that's amazing! that was a two month project and you finished it in a week!
it's because i've been listening to techno while i program
i just pretend that i'm in a computer hacking movie and it's a race against time
it's increased my productivity tenfold!
©2008 lefthandedtoons.com

http://www.lefthandedtoons.com/245/

1 Your first function [3 points]

Define a function called `sphere_surface_area` that takes the radius as a parameter and returns the surface area of a sphere with that radius.

Remember the basic structure for defining a function is:

```
def function_name(parameter1, parameter2, ...):
    statement1
    statement2
    ...
    return something # Not all functions will have return statements!
```

The way that Python can tell what is part of the function is based on the indentation.

After you've coded up your function, you can run your program and then interact with it in the Python shell by running it Python interactively from Terminal by typing:

```
% python3 -i assign2.py
```

After running this, it should leave you in the Python Shell and you can then test your functions, e.g., you could type:

```
>>> sphere_surface_area(12.4)
1932.2035136000002
```

Alternatively, you can also put in `print` statements in your code and then just run it with the run arrow. If you go this second route, make sure to remove these print statements once your confident that your function works correctly.

Once you're comfortable with running Python programs with your own defined functions and with playing with them interactively, move on to the main part of this assignment.

2 A semester abroad in Europe

http://www.bildergeschichten.eu/euro_cartoon_witz.htm

You're going to do a semester abroad in Europe and have decided to write a few functions that will help you out with some common questions you might find yourself asking while you're there.

1. [**2 points**] Write a function called `euros_to_dollars`, with a single parameter, the price in euros, and the function will return the price in dollars. Lookup the current price exchange from euros to dollars online. For example, after running your program you could type:

```
>>>  euros_to_dollars(13.5)
16.065
```

   (Note: depending on the exchange rate you use, your value will be slightly different)

   Make sure that you are using the `return` statement and not printing the answer in your function. In particular, try running the following and make sure you get something identical:

```
>>> dollars =  euros_to_dollars(13.5)
>>> dollars
15.4
```

2. [**2 points**] Write a function called `welcome` that doesn't take any parameters and returns "welcome" in some European language (English doesn't count!). For example:

```
>>> welcome()
'te lutem'
```

Remember that you can create functions with zero parameters. To call a function without any parameters, you still need to put the parenthesis at the end.

3. [**2 points**] Write a function called `kilometers_to_miles` that takes as input the number of kilometers and returns the distance in miles. For example, after running your program you could type:

```
>>> kilometers_to_miles(100)
62.137
```

4. [**1 point**] Write a function called `liters_to_gallons` that takes as input a number of liters and returns the equivalent number of gallons.

5. [**2 points**] Write a function called `mpg_from_metric` that takes *two* parameters: first the number of kilometers and second the number of liters. The function should return the miles per gallon (i.e. miles divided by gallons) by converting the kilometers and liters appropriately. *Your function must use your previous two functions:* `kilometers_to_miles` and `liters_to_gallons`. Remember, to have multiple parameters for functions, you separate them with commas.

```
>>> mpg_from_metric(400, 30)
31.358472666666668
```

6. [**2 points**] Write a function of your own that takes one or more parameters and returns something interesting/useful. Think of differences in distances, weights, volumes, speeds, recipes, etc. Brownie points for creative functions :)

7. [**2 points**] Write a function called `kilograms_to_pounds` that takes as input a weight in kilograms and returns the equivalent weight in pounds as an integer rounded down.

```
>>> kilograms_to_pounds(13)
28
```

Hint: You can use either the `//` operator or the `int` function.

8. [**3 points**] Write a function called `kgs_to_lbs_and_oz` that takes as input a weight in kilograms and returns the weight in pounds and ounces as a string. The ounces calculated should never be more than 15.

```
>>> kgs_to_lbs_and_oz(13)
'13 kilograms is 28 lbs and 10 oz.'
```

Advice

- Work through a few examples on paper to make sure you understand the calculations.
- Use an online tool to double check your answers.
- Break the calculation down into different steps and use variables to store these intermediary values.

9. [**3 points**] Write a function called `interactive_conversions` that doesn't have any parameters but instead uses the `input` function to prompt the user for input values and then displays the answer. The function should:

   - First, ask the user to enter kilometers and then print out the corresponding number of miles.
   - Then, ask the user to enter liters and print out the corresponding number of gallons.
   - Finally, ask the user to enter kilograms and print out the corresponding weight in pounds and ounces.

   Use the example transcript below as the guidelines for the exact formatting/language:

```
>>> interactive_conversions()
Enter kilometers: 100
100 kilometers is 62.137 miles
Enter liters: 2
2 liters is 0.528 gallons
Enter kilograms: 13
13 kilograms is 28 lbs and 10 oz.
```

   Note that this was an interactive run with the user manually entering 100, 2, and 13 after the corresponding prompts.

3 Testing

An important part of programming is convincing yourself that the code you are writing is correct. One way to help with this is to write test cases that check that the answer calculated by your function is the same as the answer calculated "manually". There are many ways to do this and most programming languages even have support for this built into the language, however, one easy way is just to print out and compare answers.

10. [**2 points**] Pick one of your functions above (not `euros_to_dollars` or `welcome`) and write a zero parameter test function that prints out *two test cases* for that function. A test case should include:

    - A print out of the input to the function along with the expected result (shown in blue in the example below).

- A print out of the actual result of the call to the function (shown in purple and red where the red part is the result of the actual call to the function.

Name this function, `test_name_of_function`. For example, if I were to write one for `euros_to_dollars`, it would be called `test_euros_to_dollars` and a possible output from running might be something like:

```
euros_to_dollars(200) should be:  238.0
Actual calculation:  238.0
euros_to_dollars(15) should be:  17.85
Actual calculation:  17.85
```

You may not use any of the examples in this handout as test cases (e.g., you can't use 100 as a test case for `kilometers_to_miles`).

11. [**1.5 points**] Write another zero parameter test function for one of your other function (again, not `euros_to_dollars` or `welcome`) that prints out two test cases.

12. [**0.5 points**] Add a call to each of 'your two test functions at the end of your program so that when you "run" your program (i.e., click the arrow) you'll see the results printed for both of your test functions.

4 Commenting and documentation

An important part of programming is making sure that your code is well documented so that when someone else looks at it (or, more likely, you look at it a month later) you can get information about the program without having to read all of the code.

You should have the following in your program:

- Comments at the very beginning of the file stating your name, course, assignment number and the date.
- Each function should have an appropriate docstring.
- Other miscellaneous comments to make things clear. Don't go overboard, but do include comments where things are complicated or if you have a block of code that does something. For this assignment, you won't need much since the functions should all be pretty short.

In addition, make sure that you've used good *style*. This includes:

- Following naming conventions, e.g. all variables and functions should be lowercase.
- Using good variable names.
- Proper use of whitespace, including indenting and use of blank lines to separate chunks of code that belong together.

5 Ethics

When someone says AI these days, they're often referring to "generative AI", in particular, generative large language models (LLMs). Before these language models, AI was often used synonymously with "machine learning", which is software programs that try and learn from data to make future predictions.

LLMs are being used to analyze resumes and in other ways to help with hiring. However, this is not a new use of AI. Read the following articles on AI hiring tools:

- Reuters: An article from Reuters that reports on how Amazon had to withdraw an AI recruiting tool that showed bias against women.
- Discrimination and AI hiring: A survey of the landscape of AI hiring.
- Fairness in hiring: An article looking at how AI is currently used and how that can impact fairness.

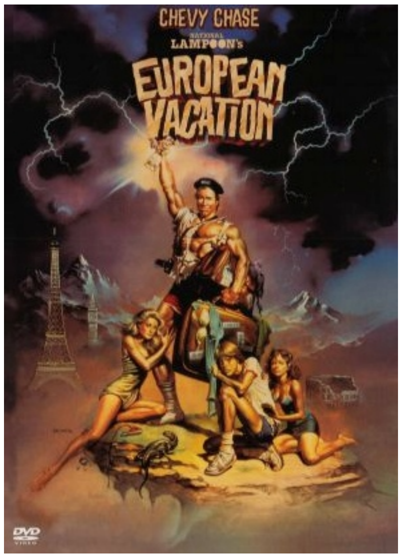
  In 1–2 paragraphs, summarize the articles and discuss anything surprising you learned, any questions, and any ideas you might have after reading the article. Export your document as a pdf file name `ethics2.pdf`.

6 When you're done

Submit your files (`assign2.py` and `ethics2.pdf`) online using Gradescope; be sure you're uploading them to the "Assignment 2" assignment. Note that you can resubmit the same assignment as many times as you would like up until the deadline.

7 Grading

	points
`sphere_surface_area`	3
A semester abroad	19
testing	4
comments and code style	3
file submitted correctly	1
ethics prompt	1
total	24 (+1)


CHEVY CHASE
NATIONAL LAMPOON'S
EUROPEAN VACATION
DVD VIDEO