

CS 1301

Pair Homework 2 – Conversions

Due: Monday September 8th, before 11:55pm

Out of 100 points

Files to submit: hw2.py

You will be writing several functions, but they will all be saved in one file: hw2.py. Please save all of the functions in this one file or you will lose points. This is a pair programming problem! You are expected to work with the person you have been paired with in class, and you are both responsible for submitting the exact same code to T-Square. Your pair may collaborate with other students in this class. Collaboration means talking through problems, assisting with debugging, explaining a concept, etc. You should not exchange code or write code for other pairs. Collaboration at a reasonable level will not result in substantially similar code. For pair programming assignments, you and your partner should turn in identical assignments with both your names at the top.

NOTE: EACH partner is responsible for turning in their own assignment! If you fail to turn in your assignment, you will receive a zero, regardless of if your partner turned in their assignment or not.

For Help:

- Visit the TA Helpdesk (schedule posted on class website)
- Email your TAs

Notes:

- **Do not forget to include the required comments and collaboration statement (as outlined on the course syllabus).**
- Do not wait until the last minute to do this assignment in case you run into problems.
- If you find a significant error in the homework assignment, please let a TA know immediately.

Conversions/Calculations (100 Points)

Section A - Queso For Fishy (10 points)

You are throwing a very big fifth birthday party for your amazingly resilient goldfish. To help celebrate, you want to spend your life's savings on queso dip. Write a function called **quesoForFishy()** that accepts one parameter which is a number representing the dollar amount of your bank account balance. Have the function **return** an **integer** representing the maximum number of queso jars you can purchase if each one costs \$2.98 and you do not exceed your bank account balance. You may ignore tax.

Sample Output:

```
>>> a = quesoForFishy(6)
>>> print( a )
2
>>> b = quesoForFishy(6.32)
>>> print( b )
2
>>> c = quesoForFishy(1234)
>>> print( c )
414
```

Section B - Mailboxes (15 points)

Studies have shown that hitting a mailbox at speeds over 30 MPH can shorten the lifespan of your car by anywhere from 2 to 6 months (not really). As a chronically bad driver, you are concerned about how much longer you have with your car. Write a function called **mailboxes()** that accepts one parameter which is an **integer** representing the number of mailboxes you have hit in your car. The function should then **print** the number of mailboxes you have hit, the least amount of months you could have removed from the lifespan of your car **and** the most amount of months based on the statistic.

Sample Output:

```
>>> mailboxes(1)
Because you have run into 1 mailbox(es), your car's life has been shortened by
anywhere from 2 to 6 months.
>>> mailboxes(5)
Because you have run into 5 mailbox(es), your car's life has been shortened by
anywhere from 10 to 30 months.
```

Section C - Concert Ticket (15 points)

You would like to go to a Taylor Swift concert, but first you want to calculate how many hours you will need to work to pay for the ticket. Write a function called **concertTicket()** that gets **input** from the user asking for the ticket price and also gets **input** from the user for their hourly rate. The function should then **print** the number of hours of work needed to earn the ticket amount. The number should be rounded to **two** decimal places.

Sample Output:

```
>>> concertTicket()
<enter 123 for the cost>
<enter 7.50 for the hourly rate>
You need to work 16.40 hours to buy your ticket
```

Section D - Boring Interlude (15 points)

You're getting tired of all of these hypothetical scenarios. Write a function called **boringInterlude()** that calculates the volume of a sphere. It should take in a **float** representing the radius of the sphere in **inches**. It should **return** the volume of the sphere in **cubic feet** as a **float**. Remember the volume of a sphere is as follows: $(4 / 3) * \pi * r^3$.

Hint: Use `math.pi` for π . Don't forget to **import math**.

Sample Output:

```
>>> a = boringInterlude(12)
>>> print( a )
4.1887902047863905
>>> b = boringInterlude(324.123)
>>> print( b )
82541.892315010977
```

Section E - Traffic Jam (15 points)

You're sitting in a traffic jam and you would like to know how many cars are in front of you at any given time. Write a function (not while you're in traffic, though) called **trafficJam()** that accepts two parameters. One should accept an **integer** representing the number of lanes, and the other should accept a **float** representing the number of miles of traffic in front of you. The function should **return** a **float** representing the number of cars in the traffic jam in front of you. Assume that each car takes up 15 feet total. You may assume the number of lanes does not change.

Hint: There are 5,280 feet in a mile.

Sample Output:

```
>>> a = trafficJam(4, 1)
>>> print( a )
1408.0
>>> b = trafficJam( 3, 3.12 )
>>> print( b )
3294.7200000000007
```

Section F - Time Left (15 points)

You are lost in the dark with a single rechargeable flashlight that is running out of battery power. Naturally, to calm your nerves, you write Python functions in your head. Write a function called **timeLeft()** that accepts a single parameter which is an **integer** representing the amount of time that you have used your flashlight in minutes. The function should then get **input** from the user asking for the number of hours that the flashlight battery lasts (this could be a **float**). The function should **print** an **integer** representing the percentage of the battery left (you can round down). The function should also **return** a **float** representing the number of minutes left before the battery dies.

Sample Output:

```
>>> a = timeLeft(30)
<enter 8 for length of time the battery lasts>
93
>>> print( a )
450.0
```

Section G - Daffodils (15 points)

Your neighbor is upset with you after you kept running over his mailbox, and now he wants you to replant some flowers that he had towards the front of his yard. His favorite color daffodils only come in groups of 12, so you will probably have some extras. Write a function called **daffodils()** that accepts the following parameters: an **integer** representing the number of flowers he needs planted, a **float** representing the price your neighbor is willing to contribute towards the flowers, and a **float** representing the price of 12 flowers. The function should **print** a sentence containing a **float** rounded to two decimal places that represents the amount of money you will have to contribute to the flowers yourself (negative values are fine).

Hint: You may find the `math.ceil()` function extremely useful for this problem. It will round a number **up** to the nearest integer. Don't forget to **import math** before using `math.ceil()`.

Example:

```
>>> math.ceil(4.1)
5.0
```

Sample Outputs:

```
>>> daffodils(20, 5.0, 10.0)
You will need to contribute $15.00.
>>> daffodils(37, 3.0, 8.0)
You will need to contribute $29.00.
>>> daffodils(37.5, 35.32, 8.34)
You will need to contribute $-1.96.
```

Rubric:

Conversions/Calculations (100 Points)

quesoForFishy() - 10 points	
Function named correctly (quesoForFishy)	5 points
Returns the correct result	5 points
mailboxes() - 15 points	
Function named correctly (mailboxes)	5 points
Calculates correct result	5 points
Prints properly	5 points
concertTicket() - 15 points	
Function named correctly (concertTicket)	5 points
Calculates correct result	5 points
Prints properly	5 points
boringInterlude() - 15 points	
Function named correctly (boringInterlude)	5 points
Calculates correct result	5 points
Returns result properly	5 points
trafficJam() - 15 points	
Function named correctly (trafficJam)	5 points
Calculates correct result	5 points
Returns result properly	5 points
timeLeft() - 15 points	
Function named correctly (timeLeft)	5 points
Properly handles a parameters and inputs	5 points
Both prints and returns correct values and types	5 points
daffodils() - 15 points	
Function named correctly (daffodils)	5 points
Uses three parameters	5 points
Prints properly with correct value	5 points