

CS 1301

Homework 1 – Python Practice & Drawing

Out of 80 points

Files to submit: 1. HW1.py

2. turtleDrawing.py

Collaboration at a reasonable level will not result in substantially similar code. Students may only collaborate with fellow students currently taking CS 1301, the TA's and the lecturer. Collaboration means talking through problems, assisting with debugging, explaining a concept, etc. **You should not exchange code or write code for others.**

For Help:

- TA Helpdesk – Schedule posted on class website.
- Email TA's or use Piazza

Notes:

- Don't 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.

Part 1 – Simple Functions

You will write a few python functions for practice with the language. In your HW1.py file, include a comment at the top with your name, section, email, and your collaboration statement. Also include each of the following functions.

1. liquidConvert
2. volumeOfCone
3. calorieCounter
4. paycheckComputation

Function Name: **liquidConvert**

Parameters: **None**

Return Value: **None**

Description:

Write a **user-interactive function** to convert any number of fluid ounces to the equivalent number of gallons, quarts, pints, and gills.

1. Get the number of fluid ounces as an integer from the user; make sure to use a descriptive prompt so the user knows what to enter.
2. Calculate the total number of gallons, quarts, pints, and gills represented by the original number of fluid ounces, using the following hints:
 - a) **There are 128 fluid ounces in a gallon.**
 - b) **There are 32 fluid ounces in a quart.**
 - c) **There are 16 fluid ounces in a pint.**
 - d) **There are 4 fluid ounces in a gill.**

The **modulo (a.k.a. remainder) operator in python is %** and will show the remainder left after an integer division. **It is useful for this problem!**

3. **Print** the calculated number of gallons, quarts, pints, and gills on one line; be sure to add appropriate labels to the display values so the user knows what the value means (e.g. if the user enters 6523 fluid ounces, you should display “6523 fluid ounces is 50 gallon(s), 3 quart(s), 1 pint(s), and 2 gill(s))

Function: **volumeOfCone**

Parameters: **None**

Return: **None**

Description:

Write a **user-interactive function** to calculate the volume of a cone

1. Get the length of the **radius in feet** from the user; make sure to use a descriptive prompt so the user knows what to enter.
2. Get the **height in feet**; again, make sure to use a descriptive prompt so the user knows what to enter.
3. Calculate the **volume of a cone** with the radius length and height entered by the user; note the volume of a cone is calculated using the formula below:

$$\text{Volume} = (\pi \times \text{Radius}^2 \times \text{Height}) / 3$$

4. **Hint: You will want to import the math module to use pi**
5. **Print** the calculated volume; be sure to add a label to the display value so the user knows what the value means (e.g. display "Volume of a cone with a radius of 7 and a height of 12 is 615.75216 feet-cubed" instead of just 615.75216)

Function: **calorieCounter**

Parameters: **None**

Return: **None**

Description:

Write a **user-interactive function** to calculate a person's calorie intake a day.

1. Get the **number of meals** a person ate from the user; make sure to use a descriptive prompt so the user knows what to enter.
2. Get the **number of miles** a person ran; make sure to use a descriptive prompt so the user knows what to enter.
3. Calculate a person's caloric intake using the following: the average calories gained **per meal is about 500 calories**, a person who has

- done **no exercise has burned about 1600 calories**, and the average calories burned **per mile of running is about 95 calories**.
4. If a person gains the same or more calories than they burn then you should print a positive number, otherwise **if a person burns more than they gain you should print a negative number**.
 5. **Print the result** in the following format: "After eating 5 meals and running 2 miles, a person gained 2500 calories and burned 1790 calories, leading to an intake of 710 calories."

Function Name: **paycheckComputation**

Parameters: **payRate, hoursWorked, taxRate**

Return Value: **payAmount**

Description: A person's pay is determined by the hours they worked times the hourly rate. Unfortunately, a percentage of their pay is taken out as taxes. Using the parameters listed above, calculate and return the amount that a person's paycheck will be for after taxes have been taken out.

1. The first parameter, **payRate**, contains a person's hourly pay.
2. The second parameter, **hoursWorked**, contains a person's number of hours worked.
3. The third parameter, **taxRate**, gives a person's take rate as a fraction between 0 and 1.
4. Calculate the person's take home pay, which is their paycheck total after taxes have been deducted.
5. **Return the result in the variable called payAmount.**

Turtles Intro

Turtles are a great way to experiment with code and view the output in a visual way. You can read a brief overview of turtles here:

http://openbookproject.net/thinkcs/python/english3e/hello_little_turtles.html

For part 2, we will be using turtles to make an image, so make sure you understand them before moving forward. You may also find this documentation useful as you work:

<https://docs.python.org/2/library/turtle.html>

Part 2 – Making the drawing (30 Points)

Files to submit: **turtleDrawing.py**

Your assignment is to make a drawing in Python using turtles. Your drawing can be of anything you want, but it must meet the following criteria:

- a) Include a recognizable non-trivial object (e.g: a house, computer, pizza, etc...)
- b) Have a background color
- c) Use penup() and pendown() in a helpful manner

When you are done, save your program as “turtleDrawing.py”. Don’t forget to include the import statement for turtles at the top with your name, section, GTID/ Email, and your collaboration statement. Try to keep things generally recognizable, while staying creative!

Grading Rubric

You will earn points as follows for each function that works correctly according to the specifications.

Part 1: Simple Functions	50 points total
liquidConvert	10
Correct function name	1
Input functions correctly written	2
Correct values are calculated	5
Printed in the correct format	2
volumeOfCone	10
Correct function name	1
Input functions correctly written	2
Correct values are calculated	5
Printed in the correct format	2
(Number of decimals doesn't matter)	
calorieCounter	15
Correct function name	1
Input functions correctly written	2
Intermediary values calculated correctly	5
Final value is correct	5
Printed in the correct format	2
(Negative if burned greater than gained)	
paycheckComputation	15
Correct function header	5
Input parameters used correctly	5
Paycheck after taxes correctly calculated	3
Result returned as payAmount	2

Part 2: turtleDrawing**30 points total**

File named correctly (turtleDrawing.py)	5
Creates a screen	5
Draws a non-trivial object	15
Sets a background color	5

You can earn up to 5 points bonus [discretion of the TAs] for extra creativity/general awesomeness, for a possible total of 85/80.