

CS 1301

Individual Homework 1 – Python Practice & Drawing

Due: Monday January 25th, before 11:55 PM

Out of 100 points

Files to submit: **1. HW1.py**
 2. treeDog.py

This is an **INDIVIDUAL** assignment!

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 T-Square Forums

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, GTID/Email, and your collaboration statement. Also include each of the following functions.

1. fluidConversion
2. coneVolume
3. calorieIntake
4. paycheckAfterTaxes

Function Name: **fluidConversion**

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:
 - There are 128 fluid ounces in a gallon.
 - There are 32 fluid ounces in a quart.
 - There are 16 fluid ounces in a pint.
 - 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: **coneVolume**

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:
Volume = (*pi* x Radius² x 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: **calorieIntake**

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: **paycheckAfterTaxes**

Parameters:

None

Return Value:

None

Description:

Write a user-interactive program to calculate a person's paycheck after taxes.

1. Ask for the total money earned from working; make sure to use a descriptive prompt so the user knows what to enter.
2. Ask for the percentage of taxes to be taken out of paycheck; again, use a descriptive prompt.
3. Calculate the person's remaining paycheck total after taxes have been deducted.
4. Print the result. The value must be formatted (both the wording and the number of decimal places) **EXACTLY** as follows: “Your corrected paycheck of \$122.25 after 25% due to taxes is \$91.69.” (Where the value for PPI represents whatever you calculated based on the inputs, and there are two digits after the decimal place. Note the period after the last digit). *You may have to use string formatting to remove extra decimal places from your calculation.*

Graphics Intro

So you're sitting there, looking at your Python window, and you find it to be, well, impersonal. So why don't we add a little personality to it? How about...a drawing?

Specifically, using the Graphics library built into Calico. The graphics library contains a multitude of commands for making graphics. See the documentation here:

http://calicoproject.org/Calico_Graphics

For this assignment, we will be focusing on the graphics aspects, so you won't need your robot. Both the link given above, as well as the PDF for the Graphics Reference will be rather helpful.

Part 2 – Making the drawing (50 Points)

Files to submit: *treeDog.py*

Your mission, should you choose to accept it (and we recommend that you do), is to make a drawing in Python that contains one or more of the following:

- a) A Tree
- b) A Dog
- c) A Tree AND a Dog
- d) Multiple Trees and Dogs

When you are done, save your program as “treeDog.py”. **Make sure that your filename matches this exactly, or you will lose points!** Don’t forget to include any necessary import statements needed to use the Graphics library or Myro functions and a comment at the top with your name, section, GTID/Email, and your collaboration statement

The drawing should contain:

- At least three different types of Calico graphics shapes/objects
- A colorful background
- A piece of text
- At least four different colors
- At least six total objects (more is better!)

Try to keep things generally recognizable (Sorry to all you budding Picassos out there). Be creative (and inoffensive)!

Grading Rubric

Grading

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

Simple Function

50 points total

fluidConversion

10

Correct function name	2
Correct values are calculated	6
Printed in the correct format	2

coneVolume

5

Correct function name	2
Prints correct value	3
(Number of decimals doesn't matter)	

calorieIntake

20

Correct function name & header	2
Gained calories correctly calculated	6
Burned calories correctly calculated	6
Intake correctly calculated	6
(Negative if burned greater than gained)	

paycheckAfterTaxes

15

Correct function name	2
Paycheck after taxes correctly calculated	8
Printed is formatted exactly as shown	5

TreeDog

50 points total

50 points

- File named correctly (treeDog.py) – 5 points
- Creates a graphics window – 5
- Draws using at least 3 types of Calico graphics shapes – 15
- Some sort of background - 5
- Uses at least 4 different colors - 10
- Use of text – 5
- Use of at least six objects - 5
- Total for drawing: 50 points

You can earn up to 3 points bonus [discretion of the TAs] for extra creativity/general awesomeness, for a possible total of 103/100.