

# CS 1301

## Individual Homework 1 – Python Practice & Snowman

Due: Friday January 16<sup>th</sup>, before 11:55 PM

Out of 100 points

---

Files to submit:      1. HW1.py  
                             2. snowman.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. Here is an example of an appropriate collaboration statement (the # character in Python is for comments that are not part of the code):

```
#George P. Burdell 902112358  
#gburdell3@gatech.edu  
#I worked worked on this assignment alone, using only this semester's course materials.
```

\*\*\*ALWAYS REMEMBER THE COLLABORATION STATEMENT\*\*\*

Also include each of the following functions.

1. convertCelsius
2. getLength
3. makeChangeUK
4. quadraticRoot

Function Name: **convertCelsius**

Parameters:

**None**

Return Value:

**None**

Description:

Write a user-interactive function to convert temperature in Celsius to temperature in Fahrenheit.

1. Get the temperature in Celsius(s) from the user; make sure to use a descriptive prompt so the user knows what to enter (e.g. “Enter the temperature in degrees Celsius”). You may assume that the user will enter a valid float (e.g. 34.4) and do not have to do error checking.
2. Convert the temperature entered by the user to degrees Fahrenheit, using the conversion factor below:

$$T_{(°F)} = T_{(°C)} \times 9/5 + 32$$

3. Print the converted temperature, be sure to add a label to the display value so the user knows what the value means (e.g. display “75.2 degrees Fahrenheit” instead of just 75.2)

Function Name: **getLength**

Parameters:

**None**

Return Value:

**None**

Description:

Write a user-interactive function to find the length of a String entered by the user.

1. Prompt the user to enter a word or phrase.
2. Find the number of total characters in the word or phrase that the user entered. (The built in Python, len() function will be useful here!)
3. Print the length of the user’s input, be sure to add a descriptive label to the display so the user knows what the value means (e.g. “The length of the input is 12 characters.”)

Function Name: **makeChangeUK**

Parameters:

**None**

Return Value:

**None**

Description:

Write a user-interactive function to convert any number of pence in British currency to the equivalent number of pounds, shillings, and pence.

1. Get the number of pence 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 pounds, shillings, and pence represented by the original number of pence, using the following hints:
  - There are 240 pence in a pound.
  - There are 12 pence in a shilling.
  - 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!
  - Also useful will be the "//" operator which is used for integer division.
3. Print the calculated number of pounds, shillings, and pence 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 1000 pence, you should display "You have 4 pounds, 3 shillings, and 4 pence", NOT just "4, 3, 4")

Function Name: **quadraticRoot**

Parameters:

**None**

Return Value:

**None**

Description:

Write a user-interactive program to find the two roots of a quadratic expression in the form of:

$$ax^2 + bx + c = 0$$

1. Prompt the user for the coefficients of quadratic, linear, and scalar terms (a, b, and c respectively in the formula above) using descriptive prompts. You may assume that valid, integer inputs will be given.
2. Using the formulae below, find both of the roots of the quadratic expression.

$$x_+ = \frac{-b + \sqrt{b^2 - 4ac}}{2a},$$

$$x_- = \frac{-b - \sqrt{b^2 - 4ac}}{2a}.$$

3. Print the results. The value must be formatted (both the wording and the number of decimal places) EXACTLY as follows: "The roots of the expression are 2.0 and 3.0." (Where the value for the roots represents whatever you calculated based on the inputs. Note the period after the last digit **without** whitespace in between). *String formatting will be helpful here. Note that we will NOT use inputs that will cause you to generate an imaginary number, so you don't have to worry about that issue!*

Example input: 1, 9, and 20 for the three prompts  
Expected output: The roots of the expression are -4.0 and -5.0.

---

## Graphics Intro

So you're sitting there, and you just can't seem to get that *Frozen* song out of your head no matter what you do, so eventually you decide to yourself: you know what? I actually am going to build a snowman!

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](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 Snowman (50 Points)

*Files to submit:*        ***snowman.py***

In order to finally get that far too catchy tune out of your head, your mission is to make a snowman in Python. You need to create a program that initializes a graphics window, and draws the snowman. When you are done, save your program as “snowman.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 snowman should contain:

- 3 stacked snowballs (for the legs, torso, and head)
- 2 eyes
- A nose (preferably a carrot one!)
- Two arms
- At least 2 different colors
- A hat (any design that combines more than one shape)

Be as creative as you want; in fact, that is encouraged! Just remember that your TA needs to be able to tell that you actually made a snowman and not just a random assortment of shapes in a window. (Sorry to all you modern art aficionados out there!)

---

## Grading Rubric

### Grading

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

#### Simple Function

**50 points total**

<b>convertCelsius</b>	<b>5</b>
<b>getLength</b>	<b>5</b>
<b>makeChangeUK</b>	<b>20</b>
<b>quadraticRoot</b>	<b>20</b>

\*more detailed breakdown below

#### Snowman

**50 points total**

50 points

- File named correctly (snowman.py) – 5 points
- Creates a graphics window – 10
- Draws 3 snowballs for the body – 5
- The snowballs are properly aligned one on top of the other – 5
- Draws 2 eyes – 5
- Draws a nose – 5
- Draws a two arms – 5
- Uses at least 2 different colors – 5
- Draws a hat by combining more than one shape – 5
- Total for Snowman: 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.*

## HW 1 Grading Rubric

Function	Total Points	Point Breakdown	
		Points	Standard
convertCelsius	5	1	Correct function header *This means the correct syntax for defining a function AND correct spelling
		1	Prompts user for input with description question
		2	Converts input from string and performs correct conversion to celsius
		1	Clear output message that describes the value's meaning
getLength	5	1	correct function header
		1	Prompts user for input with description question
		2	Correctly finds length of input
		1	Clear output message that describes the value's meaning
makeChangeUK	20	1	Correct function header
		1	Prompts user for input with description question
		1	Converts input from string
		10	Finds the correct number of pounds, shillings and pence based on input
		2	Output is on one line
		5	Output clearly describes each value's meaning
quadraticRoot	20	1	Correct function header
		4	Prompts user for two inputs and converts them appropriately
		10	Correctly calculates the quadratic roots
		5	Descriptive output formatted correctly (No space before the period!)