

Computer Science 1 (../index.html) CPSC 1301K (../index.html)

[Previous \(../modules/modulesPracticeAssignments.html\)](#)[Next \(../selection/selectionPracticeAssignments.html\)](#)

Functions Practice Assignments

Overview

Submission Instructions

For each of the following practice assignments, save your solution in a .py file, with the name being functions and the number of the assignment. For example, for "Functions01: Cups to Ounces", save your solution in a file named functions01.py (notice the lowercase "f"). Then, submit that file the respective assignment on codePost.io (<https://codePost.io>).

To register for a free account, go to <https://codepost.io/signup/join?code=I5039NQNWJ> (<https://codepost.io/signup/join?code=I5039NQNWJ>). Register with your CSU email address. Sometimes it takes more than one try, so please work on this well before the deadline. Additionally, some students have had better luck with using the "Forgot password" link.

Practice Assignments

Functions01: Cups to Ounces

Complete the cupsToOunces() function so that it takes a number of cups, converts to ounces, and returns the ounces. One cup is eight ounces.

*The main thing to note here is that you are **returning** something. There should be no print() calls within your function.*

Examples:

cupsToOunces(6) should return 48 because $6 * 8 = 48$

cupsToOunces(2.5) should return 20 because $2.5 * 8 = 20$

Provided code:

```
def cupsToOunces():  
  
# The following call will only execute when you press the  
# "run" button above (but not when you submit it).  
# You need to have cupsToOunces() return a value  
# (and not display something).  
print( cupsToOunces( 6 ) )
```

Functions02: Ounces to cups

Complete the `ouncesToCups()` function so that it takes a number of ounces, converts to cups, and returns the cups. One cup is eight ounces.

Examples:

`ouncesToCups(48)` should return 6 because $48 / 8 = 6$

`ouncesToCups(10)` should return 1.25 because $10 / 8 = 1.25$

Provided code:

```
def ouncesToCups():
```

Functions03: Reverse name

Complete the `reverseName()` function so that it takes a first name and a last name and returns a string with the entire name reversed. For example, if the first name is Sue and the last name is Walters then the function returns WaltersSue.

Note that 'takes' implies that there are parameters, in this case there are two.

Example:

`reverseName("Bob", "Smith")` returns 'SmithBob'

Provided code:

```
def reverseName():
```

Functions04: Calculate interest

Complete the `calcInterest()` function so that it takes in an interest rate per period, principal, and number of periods, and returns the amount of simple interest. Simple interest is calculated as the principal multiplied by the interest rate per period multiplied by the number of periods. The calculated interest should be expressed as a decimal.

Example:

For an interest rate of 0.01, a principal of \$10,000.00, and 10 periods, the simple interest would be \$1,000.00 *i.e.* `calcInterest(0.01, 10000.00, 12)` returns 1200.0 (because $0.01 * 10000 * 12 = 1200.0$)

Provided code:

```
def calcInterest():
```

Functions05: Property Tax

A county collects property taxes on the assessed value of property, which is 60% of the property's actual value. The property tax is then \$0.56 for each \$100 of the assessed value. Complete the `propertyTax()` function so that it takes a property's actual value and returns the property tax using 60% for the assessment value and \$0.56 per \$100 of assessment value to calculate the tax. Round to 2 decimal places. (Hint: see documentation for `round()` (<https://docs.python.org/3/library/functions.html#round>)).

For example, if a property is valued a \$50,000, it's assessment value is \$30,000 (because $50,000 * .6 = 30,000$). The tax for a property assessed at \$30,000 is \$168.00 (because $30,000/100 * .56 = 168$).

Example:

`propertyTax(50000)` returns 168.00

Provided code:

```
def propertyTax():
```

Functions06: Paint Job

A painting company has determined that for every 110 square feet of wall space, one gallon of paint and eight hours of labor will be required. The costs also depend the labor rate per hour.

Complete the `paintJobEstimator()` function so that takes the number of square feet of wall space to be painted, the cost per hour for labor, and the price of the paint per gallon. Return the total cost of the paint job. Round to two decimal places.

Note that you will use the number of square feet to determine how much paint (in gallons) and how much labor (hours) is needed. These can then be multiplied times the cost.

Example:

```
squareFeet = 200  
costPerHour = 15
```

```
pricePerGallon = 25
```

```
paintJobEstimator(squareFeet, costPerHour, pricePerGallon) returns 263.64
```

Provided code:

```
def paintJobEstimator():
```

Functions07: Stadium Seating

There are three seating categories at a stadium. For an upcoming concert, Class A seats cost \$75, Class B seats cost \$65, and Class C seats cost \$50.

Complete the `eventSeatSales()` function so that it takes in the number of seats of each class that were sold and returns the amount of income generated from ticket sales.

Example:

```
eventSeatSales(100, 50, 150) returns 18250
```

Provided code:

```
def eventSeatSales():
```

Functions08: Calories From Fat

Complete the `calFromFat()` function so that it will take a number of fat grams and return the number of calories from fat using the following calculation: calories from fat = fat grams x 9.

Round to 1 decimal place.

Example:

```
calFromFat(40) returns 360
```

Provided code:

```
def calFromFat():
```

Functions09: Calories From Carbs

Complete the `calFromCarbs()` function so that it that returns the number of calories from grams of carbohydrates according to the following calculation:

calories from carbs = carb grams * 4.

Round the result to 1 decimal place.

Example:

calFromCarbs(40.2) returns 160.8

Provided code:

```
def calFromCarbs():
```

Functions10: Division Function

Complete the divWithRem() function so that it takes two numbers, one number to be divided by the second number. The function should return the number of times the second number goes into the first and the remainder. Note that Python allows you to return more than one value. Return the values separated by a comma.

Example:

divWithRem(9, 5) should return 1 and 4

Provided code:

```
def divWithRem(x, y):
```

Functions11: Sweets Order part 1

A candy shop that sells three different types of sweets: saltwater taffy, fudge, and pralines. The shop sells the saltwater taffy for \$5.99 / lb, the fudge for \$9.99 / lb, and the pralines for \$8.99 / lb. Complete the helper functions for each type of sweet so that each takes the weight of the candy sold and returns the cost for the candy type. Round each to 2 decimal places.

Example:

costTaffy(5.5) returns 32.95

costFudge(8) returns 79.92

costPraline(2.3) returns 20.68

Provided code:

```
def costTaffy():
```

```
def costFudge():
```

```
def costPraline():
```

Functions12: Sweets Order part 2

Complete the `main()` function that uses the helper functions you wrote in part 1. The main function should get the weight of each candy type from the user, call each helper function, add the cost for each candy type, and returns the total sale.

Use the following prompts for input:

```
Enter weight of taffy in pounds:  
Enter weight of fudge in pounds:  
Enter weight of praline in pounds:
```

For the output use:

```
Your total is:
```

Note: Don't forget you need a call to `main()` at the end of the script.

Example: (user input in **bold face blue**)

```
Enter weight of taffy in pounds: 0  
Enter weight of fudge in pounds: 1.1  
Enter weight of praline in pounds: 0.9  
Your total is: 19.08
```

Provided code:

```
def costTaffy():  
    #Copy code from part 1  
  
def costFudge():  
    #Copy code from part 1  
  
def costPraline():  
    #Copy code from part 1  
  
def main():
```

Functions13: Hypotenuse part 1

Complete the function `hypotenuse()` so that it takes the lengths of the two shorter sides of a right triangle and returns the hypotenuse of the triangle, computed using the Pythagorean theorem. Round to 2 decimal places.

Note: Review the Pythagorean theorem: https://en.wikipedia.org/wiki/Pythagorean_theorem

(https://en.wikipedia.org/wiki/Pythagorean_theorem).

Example:

`hypotenuse(10, 14)` returns 17.20

Provided code:

```
def hypotenuse():
```

Functions14: Hypotenuse part 2

Complete the `main()` function that reads the lengths of the shorter sides of a right triangle from the user, uses your `hypotenuse` function to compute the length of the hypotenuse, and displays the result.

Use the following prompts for user input:

```
Enter side 1:
Enter side 2:
```

Use the following for output to the screen:

```
The hypotenuse is:
```

Note: Don't forget you need a call to `main()` at the end of the script.

Provided code:

```
# Copy hypotenuse function from part 1

def main():
```

Functions15: Taxi Fare part 1

Complete the `taxiFare()` function so that it takes a base rate, a rate per mile, and the number of miles and returns the total taxi fare rounded to 2 decimal places. The total fare is the base rate plus the rate per miles times the number of miles.

Example:

`taxiFare(3.15, 1.20, 30)` returns 39.15

Provided code:

```
def taxiFare():
```

Functions16: Taxi Fare part 2

Complete a `main()` function so that it takes input from the user to get a base rate, rate per mile, and number of miles traveled and then uses your `taxiFare()` function to get the total fare. Lastly, it should print the total fare for the user.

Use the following prompts for input:

```
Enter base rate:  
Enter cost per mile:  
Enter number of miles traveled:
```

Use the following for output:

```
Your fare is:
```

Note: Don't forget you need a call to `main()` at the end of the script.

Provided code:

```
# Copy taxiFare() from from Part 1  
  
def main():
```

Functions17: Turtle Mural part 1

You are interested in painting a mural that you designed with turtle using only a circle, a rectangle, and a triangle. You need to calculate the area that will need to be painted. Complete 3 helper functions (1 for each shape) that take the needed information, calculate, and return the area. Round each to 2 decimal places. For the triangle, have it take the base and the height of the triangle. For the rectangle, have it take the length and the width. For the circle, have it take the radius.

Note 1: If you need a reminder of the formulas for area check: <https://en.wikipedia.org/wiki/Area> (<https://en.wikipedia.org/wiki/Area>).

Note 2: Import `math` and use the constant `pi` for a more accurate value of `pi`

Examples:

`areaTriangle(12, 14.4)` returns 86.4

`areaRectangle(13.2, 4.5)` returns 59.4

`areaCircle(52.1)` returns 8527.57

Provided code:


```
def areaTriangle():  
  
def areaRectangle():  
  
def areaCircle():
```

Functions18: Turtle Mural part 2

Complete a `main()` function to get input from the user about the size of the triangle, rectangle, and circle in the mural. Use the helper functions you wrote in part 1 to calculate the areas of each shape. Add these together in `main()` to get the total area. Lastly, display the total area, rounded to 2 decimal places.

Use the following prompts for input:

```
Enter the base of the triangle:  
Enter the height of the triangle:  
Enter the length of the rectangle:  
Enter the width of the rectangle:  
Enter the radius of the circle:
```

Use the following for output:

```
Total area is:
```

Note: Don't forget you need a call to `main()` at the end of the script.

Provided code:

```
# Copy areaTriangle() from part 1  
  
# Copy areaRectangle() from part 1  
  
# Copy areaCircle() from part 1  
  
def main():
```

Functions19: CSU Taco Shop

Good code is organized into functions. This assignment allows you to demonstrate organizing your code into functions (including a `main()` function).

Write a program (with comments) that take an order of five different tacos at a taco shop.

You should have a helper function that prints a welcome message and the menu.

You should have another helper function for **each** item that a user can order that takes in the quantity of that item as an argument and returns the total cost (to 2 decimal places), where the follows are the prices.

Barbeque beef taco	\$2.90
Shrimp and grits taco	\$3.00
Cajun fish taco	\$3.25
Chicken guacamole taco	\$2.75
Caramelized onion lentil taco	\$2.50

These helper functions should not print nor request user input.

You should have a `main()` function that asks the user for the quantity of each of the five tacos and calculates the total cost of the order.

The `main()` function should also calculate the total with sales tax (8%). (The total price with sales tax is calculated by multiplying the total before tax by 1.08).

The `main()` function should print the user's order as well as the total cost of the order before and after sales tax (all to at most 2 decimal places).

Example (user input highlighted in gray and in **bold text**):

```
Welcome to the CSU Taco Shop!
Menu:
Barbeque beef taco           $2.90
Shrimp and grits taco        $3.00
Cajun fish taco              $3.25
Chicken guacamole taco       $2.75
Caramelized onion lentil taco $2.50
How many barbeque beef tacos would you like? 0
How many shrimp and grits tacos would you like? 1
How many cajun fish tacos would you like? 2
How many chicken guacamole tacos would you like? 3
How many caramelized onion lentil tacos would you like? 4
Your order is:
0 Barbeque beef tacos
1 Shrimp and grits tacos
2 Cajun fish tacos
3 Chicken guacamole tacos
4 Caramelized onion lentil tacos
Your total before tax is $27.75
Your total with tax is $29.97
```

Provided code:

```
def beef_taco():  
    return  
  
def shrimp_taco():  
    return  
  
def fish_taco():  
    return  
  
def chicken_taco():  
    return  
  
def lentil_taco():  
    return  
  
# Write a main() function that gets user input, utilizes all of the above working  
# helper functions, and calculates total price before and after tax.
```