|  |  |
| --- | --- |
| **NAME** | **DUE** |
| **Tomoki Koike** | **2/23/2019** |

**Every Boiler Engineering Code – Entry Level Programming**

**Week 1 – Programming Exercises**

1. A cookie recipe calls for the following ingredients:
   * 1.5 cups of sugar
   * 1 cup of butter
   * 2.75 cups of flour

The recipe produces 48 cookies with this amount of the ingredients. Write a program that asks the user how many cookies he or she wants to make, then displays the number of cups of each ingredient needed for the specified number of cookies. **(Note: the precision of the number output must be set to 2, which means the number output should be rounded to two decimal places)**

**(Use input 100 to test your program, then take screenshot of the outputs.)**

**Code:**

##

# Author: Tomoki Koike

# Date: 2/26/2019

# Description: (HW1 Q1)

# This program is designed to calculate the required amounts of ingredients for

# a certain cookie recipe depending on the user input (without input validation)

# Student of 2021

##

# The cups of each ingredients necessary for 48 cookies

sugar = 1.5

butter = 1

flour = 2.75

# Number of cups necessary to make 1 cookie

sugar\_per\_ck = sugar / 48

butter\_per\_ck = butter / 48

flour\_per\_ck = flour / 48

# User input of the number of cookies to make

num\_cookies = int(input('How many cookies do you want to make? -> '))

# Calculating the amount of each ingredients required

sugar\_req = num\_cookies \* sugar\_per\_ck

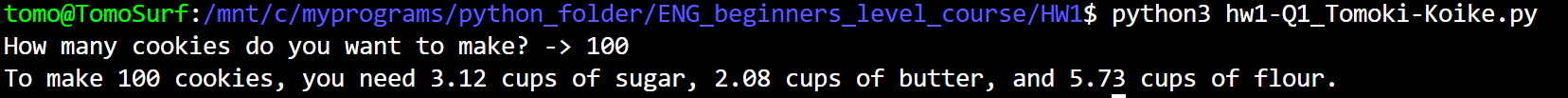
butter\_req = num\_cookies \* butter\_per\_ck

flour\_req = num\_cookies \* flour\_per\_ck

# Output

print('To make', num\_cookies, 'cookies, you need', '{0:.2f}'.format(sugar\_req), 'cups of sugar,', '{0:.2f}'.format(butter\_req), 'cups of butter, and', '{0:.2f}'.format(flour\_req), 'cups of flour.')

**Sample Execution:**



1. A vineyard owner is planting several new rows of grapevines, and needs to know how many grapevines to plant in each row. She has determined that after measuring the length of a future row, she can use the following formula to calculate the number of vines that will fit in the row, along with the trellis end-post assemblies that will need to be constructed at each end of the row:

***V = (R-2E) / S***

The terms in the formula are:

***V*** is the number of grapevines that will fit in the row.

***R*** is the length of the row, in feet.

***E*** is the amount of space, in feet, used by an end-post assembly.

***S*** is the space between vines, in feet.

Write a program that makes the calculation for the vineyard owner. The program should ask the user to input the following:

* The length of the row, in feet
* The amount of space used by an end-post assembly, in feet
* The amount of space between the vines, in feet

Once the input data has been entered, the program should calculate and display the number of grapevines that will fit in the row. (**(Note: the output must be formatted to an integer, e. g. if there are 33.5 vines that will fit in the row, your program’s output should be 33)**

**(Use R as 60.0, E as 2.0, S as 2.5 to test your program, then take screenshot of the outputs)**

**Code:**

##

# Author: Tomoki Koike

# Date: 2/26/2019

# Description: (HW1 Q2)

# This program is designed to compute the number of grapevines possible to

# fit in each row of a vineyard depending on several parameters (with input validations)

# Student of 2021

##

# Accepting user inputs (parsing through validation for each input)

# Input length of one row

while True:

try:

row\_length = float(input('What is the length (ft) of the rows? -> '))

except ValueError:

print('Sorry, cannot understand that. Please enter again.')

continue

if row\_length <= 2:

print('Please enter a value larger than 3.')

continue

else:

# Valid input

break

# Input amount of space used by the post-end assembly

while True:

try:

assm\_space = float(input('What is the amount of space (ft) used by the post-end assembly? -> '))

except ValueError:

print('Sorry, cannot understand that. Please enter again.')

continue

if assm\_space > 0.5 \* row\_length:

print("Please enter value smaller than half of the row length.")

continue

else:

# Valid input

break

# Input the space between the vines

vine\_space = float(input('What is the space (ft) between the vines? -> '))

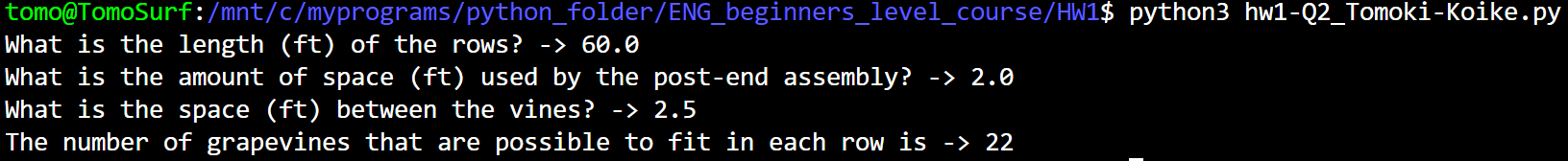
# Calculating the number of grapevines that fit in each row

row\_fit = int((row\_length - 2 \* assm\_space) / vine\_space)

# Output

print('The number of grapevines that are possible to fit in each row is ->', row\_fit)

**Sample Execution:**



1. When a bank account pays compound interest, it pays interest not only on the principal amount that deposited into the account, but also on the interest that has accumulated over time. Suppose you want to deposit some money into a savings account, and let the account earn compound interest for a certain number of years. The formula for calculating the balance of the account after a specified number of years is:

***A = P (1 + r/n)nt***

The terms in the formula are:

***A*** is the amount of money in the account after the specified number of years.

***P*** is the principal amount that was originally deposited into the account.

***r*** is the annual interest rate.

***n*** is the number of times per year that the interest is compounded.

***t*** is the specified number of years.

Write a program that makes the calculation for you. The program should ask the user to input the following:

* The amount of principal originally deposited into the account
* The annual interest rate paid by the account
* The number of times per year that the interest is compound (For example, if interest is compounded monthly, enter 12. If interest is compound quarterly, enter 4)
* The number of years the account will be left to earn interest

Once the input data has been entered, the program should calculate and display the amount of money that will be in the account after the specified number of years.

**(Note: the user should enter the interest rate as a percentage. For example, 2 percent would be entered as 2, not as .02. The program will then have to divide the input by 100 to move the decimal point to the current position. And the precision of output must be set to 2, the output must be formatted with comma separators and with the ‘$’ sign. For example, the result of your program is 357689.237, then the output should be formatted as $357,689.24)**

**(Use P as 10000.00, r as 4.9, n as 4, t as 3 to test your program, then take screenshot of the outputs)**

**Code:**

##

# Author: Tomoki Koike

# Date: 2/26/2019

# Description: (HW1 Q3)

# This program is designed to compute he balance after specified years from deposit considering the compound interest (with input validations)

# Student of 2021

##

# Function to validate input values to be positive numbers

def get\_positive\_input(prompt):

while True:

try:

value = float(input(prompt))

except ValueError:

print('Sorry, cannot understand. Please enter again.')

continue

if value > 0:

# valid input

break

else:

print('Please enter a positive value.')

continue

return value

# Accepting user inputs

# Principle amount orginally deposited into account

pAmount = get\_positive\_input('Please enter the principle amount -> ')

# Interest rate

intrstRate = get\_positive\_input('Please enter the interest rate -> ')

# Number of times per year that the interest is compounded

perYear\_compound = get\_positive\_input('Please enter the number of times per year that the interest is compounded -> ')

# Number of years the account will be left to earn interest

num\_years = get\_positive\_input("Please enter the number of years that the account will be left to earn interest -> ")

# Convert percentage of interest rate to decimals

intrstRate = intrstRate / 100

# Calculate the total balance

balance = pAmount \* (1 + intrstRate / perYear\_compound)\*\*(perYear\_compound \* num\_years)

# Output

print('The balance in the account after', int(num\_years), 'years is', '{0:.2f}'.format(balance))

**Sample Execution:**

