**Fall 2019 EBEC – Entry Level Programming in Python**

**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 the following numbers to test:**

|  |  |
| --- | --- |
| **Input** | **Correct Output** |
| **48** | **1.50 cups of sugar**  **1.00 cups of butter**  **2.75 cups of flour** |
| **100** | **3.12 cups of sugar**  **2.08 cups of butter**  **5.73 cups of flour** |
| **233** | **7.28 cups of sugar**  **4.85 cups of butter**  **13.35 cups of flour** |

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 the following numbers to test:**

|  |  |
| --- | --- |
| **Input** | **Correct Output** |
| **R=55**  **E=2.5**  **S=2** | **You have enough space for 25 vine(s).** |
| **R=2000**  **E=12**  **S=35** | **You have enough space for 56 vine(s).** |
| **R=15**  **E=3**  **S=5** | **You have enough space for 1 vine(s).** |

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, in percent
* 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)**

**Use the following numbers to test:**

|  |  |
| --- | --- |
| **Input** | **Correct Output** |
| **P=10000**  **r=4.9**  **n=4**  **t=3** | **At the end of 3 years you will have $ 11,573.20** |
| **P=80000**  **r=0.25**  **n=12**  **t=1.5** | **At the end of 1.5 years you will have $ 80,300.53** |
| **P=10000000**  **r=0.25**  **n=1**  **t=1.5** | **At the end of 1.5 years you will have $ 10,037,523.43** |