|  |  |  |  |
| --- | --- | --- | --- |
| **Instructor** |  | **Due Date** |  |

**PROJECT Loops and Repetition - Bank Account Activity 50 points**

**Objective** To write a program that will show bank account balances over time.

***PROJECT DESCRIPTION***

Write a program that will prompt a user for an initial bank account balance and an annual interest rate, which the bank allows for their clients.

Once the user has supplied the initial deposit amount and interest rate for the account, your program should then proceed to calculate the new balance for the account  
 ( including interest earned ) over a twelve - month period ( taking into account the additional amount the interest earned for each month ) . A simple way of handling the calculation of the balance for each month of the investment period is shown below as an example.

**balance = balance + monthly deposit amount**

**interest = balance ( current ) × ( interest rate / 12 )**

**balance ( new ) = balance ( current ) + interest**

Note that the interest rate is per annum (across the entire year), so the interest rate percentage for each month is equivalent to the yearly interest rate divided by 12 .   
 Your program should then display the new balance calculated for each month for a total of 12 months. A sample snapshot of this follows in **Figure 1** below.

***Information about This Project***

Here is an example of the above programming scenario:

Consider an initial bank balance of $ 2,000 and an annual interest rate of 4 % and the first three months of account activity.

Month 1 ( ending balance ) $ 2,000 + ( 4 % / 12 ) × $ 2,000 = $ 2,006.67

Month 2 ( ending balance ) $ 2,006.67 + ( 4 % / 12 ) × $ 2,006.67 = $ 2,013.36

Month 3 ( ending balance ) $ 2,013.36 + ( 4 % / 12 ) × $ 2,013.36 = $ 2,020.07

Here is some information and examples on some looping structures that can be useful when writing your program statements for this project.

**[ for Loop in Python ]**

**start = 1**

**stop = 5**

**for count in range(start, stop + 1) :**

**# start loop body**

**print ("iteration number: ", count)**

**# end loop body**

**[ Output ]**

**iteration number: 1**

**iteration number: 2**

**iteration number: 3**

**iteration number: 4**

**iteration number: 5**

**PROJECT Loops and Repetition - Bank Account Activity**

**[ While Loop in Python ]**

**count = 1**

**while (count < 5) :**

**# start loop body**

**print ("the count is: ", count)**

**count = count + 1**

**# end loop body**

**[ Output ]**

**the count is: 1**

**the count is: 2**

**the count is: 3**

**the count is: 4**

**[ Repetitive Program Control ]**

The three types of program control include sequential, selection and repetition. Repetitive program control arises when the execution of some programming statements occurs in a repetitive or looping fashion.

***Steps to Complete This Project***

**STEP 1**  **Open the Python IDLE IDE and Write the Program Code**

Open the Python IDLE IDE ( Integrated Development Environment ) or similar Python development environment on your computer.

You will notice when you initially open Python, the default is an interpretive shell allowing only for single commands to be given. You really need to enter in a whole program then execute it to work any of the labs for the course. To start entering code into IDLE go to **File > New File** from your menu. This will allow you to enter your source code in an editor style format like Notepad.

**STEP 2**  **View the Sample Program Run**

**Figure 1** shows a sample program run of this project ( prior to the project modification ) . Study the input and output values of the sample program run.

The sample program run will assist you in understanding the required code statements that are necessary to complete this project.

**PROJECT Loops and Repetition - Bank Account Activity**

**Figure 1 Sample Program Run ( Prior to the Project Modification )**

|  |
| --- |
| **>>>**  **Enter an initial bank balance 2000**  **Include annual interest rate ( as a decimal ) 0.04**  **Month: 1 New Monthly bal: $2006.67**  **Month: 2 New Monthly bal: $2013.36**  **Month: 3 New Monthly bal: $2020.07**  **Month: 4 New Monthly bal: $2026.80**  **Month: 5 New Monthly bal: $2033.56**  **Month: 6 New Monthly bal: $2040.33**  **Month: 7 New Monthly bal: $2047.14**  **Month: 8 New Monthly bal: $2053.96**  **Month: 9 New Monthly bal: $2060.81**  **Month: 10 New Monthly bal: $2067.68**  **Month: 11 New Monthly bal: $2074.57**  **Month: 12 New Monthly bal: $2081.48**  **>>>** |

**STEP 3 Modify Your Program Code**

Once your program is fully working, modify your program by creating a loop that will allow a user 3 tries to enter in a pin number, which if correct, will allow the user to enter information such as their initial balance and desired annual interest rate as demonstrated above.

Include, in your code, an error message if an incorrect pin number was entered in and state how many tries the user has left to correctly input their pin number to gain access to the rest of the program.

Show a snapshot of your running program where a person enters an incorrect pin as well as the correct pin (pin has to be some alphanumeric number at least 4 digits in length) along with the inputs for an initial balance and an annual interest rate and resulting monthly displays of new balances.

**STEP 4**  **Submit Your Program**

When completed, your program source code (.py file) as well as include

snapshots of your running program input/output results along with your name

and course number, date/time etc. into a MS Word document. Submit your files

to the appropriate course submittal box when complete.

\*Grads include a secondary loop which will allow for users to enter in

additional attempts to enter in an initial balance and an interest rate. Allow for

some sentinel value to stop user input and cause the program to exit.

Show a snapshot of an additional attempt showing an additional balance

entered by the user as well as the program being successfully terminated.