CIS 106 – Loops Part 2

For each problem prepare an IPO chart. Then write the code for each. Save the IPO within this document and upload to your repository. After code is complete upload the files (.py) to your repository. Paste the link to your repository into the assignment completion link in Blackboard.

1. Allow the user to enter a principle amount and interest rate repeatedly (need a loop to control the program execution). Compute the annual interest (principle x rate). Compute ending balance to be principle (beginning balance + interest). Display year, beginning balance and ending balance for each of the 5 years. Display the accumulated interest for the 5 years. Note: the new balance by year (this will be the principle for the following year. Format the output.

Example:

Enter principle amount: 10000.00

Enter interest rate: 0.10

Year Beginning Ending

Balance Balance

1 $10,000.00 $11,000.00

2 $11,000.00 $12,100.00

3 $12,100.00 $13,310.00

4 $13,310.00 $14,641.00

5 $14,641.00 $16,105.00

Total interest earned: $6,156.00

| input | output | process |
| --- | --- | --- |
| Principle amount (float) | Initialize  ‘total\_interest’ to 0 | Display “enter  Principle amount” |
| Interest rate (float) | For each year from 1 to 5: | Display “enter interest rate (as a decimal)” |
|  | Calculate interest for the year as ‘principle \* interest\_rate’ | Display the year beginning balance, and ending balance for each ending balance for each year in a formatted table |
|  | Calculate ending balance as ‘principle + interest’ | Display the total interest earned over the 5 years |
|  | Add the interest to  ‘total\_interest’ |  |
|  | Update ‘principle’ to  The new ending balance for the next iteration |  |
|  | Print the results in a formatted table including year beginning balance and ending balance |  |
|  | Print the total interest earned over the 5 years |  |

2. Fibonacci sequence is a sequence of natural order. The sequence is:

1, 1, 2, 3, 5, 8 etc

Use of for loop compute and display first 20 numbers in the sequence. Hint: start with 1 , 1.

| input | process | Output |
| --- | --- | --- |
|  | Initialize the first two numbers in the sequences: ‘a’ and ‘b’ both set 1 | Display the 20 first #’s in the Fibonacci sequence |
|  | Create a list ‘sequence’ to store the sequence and initialize it with the first two numbers |  |
|  | Loop 18 times (since the first two numbers are already in list) |  |
|  | In each iteration: |  |
|  | Calculate the next number in the sequence as the sum of ‘a’ and ‘b’ |  |
|  | Update ‘a’ to be ‘b’ and ‘b’ to be the new sum |  |
|  | Append the new number to the ‘sequence’ list |  |
|  | Print the sequence |  |
|  | Format the output to display numbers separated by commas, removing the trailing comma and space at the end |  |

3. Create a text file that contains employee last name and salary. Read in this data. Determine the bonus rate based on the chart below. Use that rate to compute bonus. For each line display the employee last name, salary and bonus. After the loop display the sum of all bonuses paid out.

Salary Bonus Rate

100,000.00 and up 20%

50,000.00 15%

All other salaries 10%

Example file (create your own data with at least 5 lines:

Adams

50000.00

Baker

75000.00

Smith

45000.00

Etc

| Input | proccess | output |
| --- | --- | --- |
| Text file containing employee last name and salary | Open and read the text file | Display each employee’s last name salary and computed bonus |
|  | Parse the data into a list of tuples | Display the total sum of all bonuses paid out |
|  | Initialize ‘total\_bonus’ to 0 |  |
|  | Loop through the list of employee data |  |
|  | For each employee determine the bonus rate based on salary |  |
|  | Compute the bonus using the ‘calculate\_bonus’ function |  |
|  | Add the computed bonus to ‘total\_bonus’ |  |
|  | Print the emplotee’s last name salary and bonus |  |
|  | After the loop print the total sum of all bonuses |  |

4. Create a text file with item, quantity and price. Read through the file one line at a time. Compute the extended price (quantity x price). For each line display the item, quantity, price and extended price. After the loop display the sum of all the extended prices, the count of the number of orders and the average order.

Example Data File

Widget

10

50

Hammer

2

10

Saw

4

8

Etc

| Input | process | Output |
| --- | --- | --- |
| ‘item.txt’ file | Open and read the file | Lines from the file read into lists |
|  | Initialize lists for items, quantities and prices | Lists populated with items, quantities, and price |
|  | Loop through lines in the file  Extracting item, quantity and price in groups of three. | Each item details stored in separate lists |
|  | Calc extended prices for each item (quantity times price) | Extended prices calculated for each item |
|  | Sum the extended price | Total extended prices calculated |
|  | Count the number of orders | Number of orders counted |
|  | Calc the average order value (total extended price / number of orders) | Average order value calculated |
|  | Print each item’s details (item, quantity, price, extended price) | Detail of each item shown |
|  | Print the total extended price number of orders and average order value | Total extended price, number of orders and average order value shown. |

5. Create a text file with student last name, district code (I or O) and number of credits taken. Compute tuition owed (credits taken x cost per credit). Cost per credit for in district students (district code I) is 250.00. Out of district students pay 500.00 per credit. For each line display student last name, credits taken and tuition owed. After the loop display sum of all tuition owed and the number of students.

Example file

Jones

I

12

Adams

I

10

Baker

O

12

Smith

O

16

| Input | process | Output |
| --- | --- | --- |
| ‘students.txt’ | Open and read ‘students.txt’ file | Student last name, credits taken, tuition owed for each student |
| (each line contains last name, district code, and credits taken, separated by commas) | For each line | Total tuition owed and # of students |
|  | Split the line by commas to extract last name, district code, and credits taken |  |
|  | Convert ‘credits taken’ to an integer |  |
|  | Determine cost per credit based on the district code (‘I’ or ‘O’) |  |
|  | Calculate tuition owed (‘credit taken’ \* ‘cost per credit’) |  |
|  | Print student’s last name, credits taken , and tuition owed |  |
|  | Add tuition owed to the total tuition |  |
|  | Increment the student count. |  |
|  | After the loop, print the total tuition owed and number of students. |  |