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.

|  |  |  |
| --- | --- | --- |
| Input | Process | Output |
| User enters principal amount (initial investment) | Validate the entered principal amount and interest rate | Display year, beginning balance, interest, and ending balance for each year |
| User enters interest rate (in percentage) | Initialize principal and rate. | Display the total accumulated interest for 5 years |
|  | Use a loop to calculate interest for 5 years: |  |
|  | For each year, calculate annual interest: `interest = principal \* rate` |  |
|  | - Calculate ending balance: `ending\_balance = principal + interest` |  |
|  | Update principal for the next year with the ending balance |  |
|  | Accumulate the total interest over the 5 years |  |
|  | Display results: format output for each year |  |

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

1. 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 |
| Initial Fibonacci sequence: 1, 1 | Set first two numbers in the sequence as `a = 1, b = 1` | Display Fibonacci sequence |
| Fixed sequence length: 20 | Use a for loop to calculate next Fibonacci number using: `next\_num = a + b` | Display each Fibonacci number |
|  | Update sequence: `a = b` and `b = next\_num` | Final list of 20 numbers |
|  | Repeat steps until 20 numbers are computed. |  |
|  |  |  |

1. 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%

|  |  |  |
| --- | --- | --- |
| Input | Process | Output |
|  |  |  |
| Employee Name, Salary | Read the file line by line |  |
| Salary |  |  |
|  |  |  |
|  |  |  |

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

Adams

50000.00

Baker

75000.00

Smith

45000.00

Etc

1. 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.

|  |  |  |
| --- | --- | --- |
| Input | Process | Outputs |
|  |  |  |
| Item | Read file | Item (displayed on screen) |
| Quantity | Compute Ext. Price | Quantity ( displayed on screen) |
| Price | ( Quantity \* Price) | Price ( displayed on screen) |
|  | Display item, quantity,  Price, and ext. price | Ext. Price |
|  | Sum all ext prices | Sum of all Ext. prices |
|  | Count number of orders | Count of orders |
|  | Compute average order | Average Order |

Example Data File

Widget

10

50

Hammer

2

10

Saw

4

8

Etc

1. 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.

|  |  |  |
| --- | --- | --- |
| Inputs | Process | Outputs |
|  | Totaltuition = 0  C = 0 |  |
| lastname | Get first lastname |  |
| dcode | While not at end  Get dcode, credits  If dcode = ‘I’  Costpercredit = 250  Else  Costpercredit = 500  Tuition = costpercredit \* credits  C = c+ 1  Totaltuition = totaltuition + tuition  Display tuition , credits, tuition  Get next lastname | Lastname  Credits  Tuition  Totaltuition  c |
| credits |  |  |
|  | Display totaltuition  Display c |  |
|  |  |  |
|  |  |  |

Example file

Jones

I

12

Adams

I

10

Baker

O

12

Smith

O

16