CIS 106 – Loops Part 2 - Tirth Patel

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.

* 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

IPO CHART

Input:

- Principal amount (initial amount of money)

- Annual interest rate

Processing:

- Using a loop, calculate the annual interest and ending balance for each of the 5 years.

- Accumulate the interest for each year.

- Update the principal amount for the following year.

Output:

- Display year, beginning balance, and ending balance for each of the 5 years.

- Display the accumulated interest for the 5 years.

Year Beginning Ending

Balance Balance

* $10,000.00 $11,000.00
* $11,000.00 $12,100.00
* $12,100.00 $13,310.00
* $13,310.00 $14,641.00
* $14,641.00 $16,105.00

Total interest earned: $6,156.00

--CODE DONE

* 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:

- None (No user input required)

Processing:

- Initialize the first two Fibonacci numbers (1, 1).

- Use a for loop to calculate the next 18 numbers in the sequence.

- In each iteration, add the last two numbers to get the next number.

- Store and display the numbers in the sequence.

Output:

- Display the first 20 numbers in the Fibonacci sequence.

--CODE DONE

* 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.

Input:

- Data from a text file containing employee last name and salary.

- Bonus rate chart (mapping of salary ranges to bonus rates).

Processing:

- Read employee data from the text file.

- Determine the bonus rate for each employee based on the salary and the provided bonus rate chart.

- Calculate the bonus for each employee (bonus = salary \* bonus rate).

- Display the employee's last name, salary, and bonus for each employee.

- Accumulate the bonuses to calculate the total bonus paid out.

Output:

- Display the employee's last name, salary, and bonus for each employee.

- 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

--CODE DONE

* 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:

- Data from a text file containing item, quantity, and price for each line.

Processing:

- Read the data from the text file one line at a time.

- For each line, parse the item, quantity, and price values.

- Calculate the extended price for each line (extended price = quantity \* price).

- Display the item, quantity, price, and extended price for each line.

- Accumulate the extended prices and count the number of orders.

Output:

- Display the item, quantity, price, and extended price for each line.

- Display the sum of all extended prices.

- Display the count of the number of orders.

- Calculate and display the average order.

Example Data File

Widget

10

50

Hammer

2

10

Saw

4

8

Etc

--CODE DONE

* 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.

Input:

- Data from a text file containing student last name, district code (I or O), and number of credits taken for each line.

Processing:

- Read the data from the text file one line at a time.

- For each line, parse the student's last name, district code, and number of credits.

- Calculate the tuition owed based on the district code (in-district: $250 per credit, out-of-district: $500 per credit).

- Display the student's last name, credits taken, and tuition owed for each line.

- Accumulate the total tuition owed and count the number of students.

Output:

- Display the student's last name, credits taken, and tuition owed for each line.

- Display the sum of all tuition owed.

- Display the number of students.

Example file

Jones

I

12

Adams

I

10

Baker

O

12

Smith

O

16

--CODE DONE