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 | Process | Output |
| Principle amount | |  | | --- | | - For each year (from 1 to 5): |  |  | | --- | |  | | |  | | --- | | - Display year number, beginning balance, and ending balance |  |  | | --- | |  | |
| Interest Rate | |  | | --- | | - Compute annual interest: annual\_interest = principal \* interest\_rate |  |  | | --- | |  | | |  | | --- | | - Accumulated interest over 5 years |  |  | | --- | |  | |
| Loop Control for 5 years | |  | | --- | | - Compute ending balance for the year: ending\_balance = principal + annual\_interest |  |  | | --- | |  | |  |
|  | |  | | --- | | - Display the current year, beginning balance, and ending balance |  |  | | --- | |  | |  |
|  | |  | | --- | | - Update the principal (beginning balance for the next year) with the ending balance |  |  | | --- | |  | |  |
|  | |  | | --- | | - Accumulate interest over the 5 years for final output |  |  | | --- | |  | |  |
|  |  |  |

1. Display the odd numbers starting at 1 and ending with 25. Use a for this problem.

|  |  |  |
| --- | --- | --- |
| Input | Process | Output |
|  | Use a for loop to specify the numbers between 1 and 25 | Display odd numbers between 1 and 25 |
|  | |  | | --- | | - Check if the current number in the loop is odd |  |  | | --- | |  | |  |
|  | |  | | --- | | - If the number is odd, print it. |  |  | | --- | |  | |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

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%

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

Adams

50000.00

Baker

75000.00

Smith

45000.00

Etc

|  |  |  |
| --- | --- | --- |
| Input | Process | Output |
| |  | | --- | | - Employee last names and salaries from a text file. |  |  | | --- | |  | | |  | | --- | | - Open the file containing employee data (last name and salary). |  |  | | --- | |  | | |  | | --- | | - Display employee last name, salary, and computed bonus. |  |  | | --- | |  | |
| |  | | --- | | - Salary ranges and their corresponding bonus rates. |  |  | | --- | |  | | |  | | --- | | - For each employee, determine their bonus rate based on their salary using the provided chart. |  |  | | --- | |  | | |  | | --- | | - Display the total sum of all bonuses paid out at the end. |  |  | | --- | |  | |
|  | |  | | --- | | - Calculate the bonus using the formula: bonus = salary \* bonus\_rate. |  |  | | --- | |  | |  |
|  | |  | | --- | | - Accumulate the total bonus paid out. |  |  | | --- | |  | |  |
|  | |  | | --- | | - Display the last name, salary, and computed bonus for each employee. |  |  | | --- | |  | |  |
|  | |  | | --- | | - After the loop, display the total sum of bonuses. |  |  | | --- | |  | |  |
|  |  |  |

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.

Example Data File

Widget

10

50

Hammer

2

10

Saw

4

8

Etc

|  |  |  |
| --- | --- | --- |
| Input | Process | Output |
| |  | | --- | | - Text file containing item, quantity, and price for each order. |  |  | | --- | |  | | |  | | --- | | - Open and read the file line by line. |  |  | | --- | |  | | |  | | --- | | - Display each item, quantity, price, and extended price. |  |  | | --- | |  | |
| |  | | --- | | - For each line, extract the item name, quantity, and price. |  |  | | --- | |  | | |  | | --- | | - For each line, compute the extended price using the formula: extended price = quantity \* price. |  |  | | --- | |  | | |  | | --- | | - After the loop, display: |  |  | | --- | |  | |
|  | |  | | --- | | - Keep track of the total sum of extended prices. |  |  | | --- | |  | | |  | | --- | | - Total sum of all extended prices. |  |  | | --- | |  | |
|  | |  | | --- | | - Count the number of orders processed. |  |  | | --- | |  | | |  | | --- | | - Total number of orders processed. |  |  | | --- | |  | |
|  | |  | | --- | | - After processing all lines, compute the average order: average order = total extended price / number of orders. |  |  | | --- | |  | | Average order amount |
|  |  |  |
|  |  |  |

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.

Example file

Jones

I

12

Adams

I

10

Baker

O

12

Smith

O

16

|  |  |  |
| --- | --- | --- |
| Input | Process | Output |
| |  | | --- | | - Text file containing student last names, district codes, and the number of credits taken. |  |  | | --- | |  | | |  | | --- | | - Open and read the file line by line. |  |  | | --- | |  | | |  | | --- | | - Display each student's last name, credits taken, and tuition owed. |  |  | | --- | |  | |
| |  | | --- | | - Extract last name, district code, and number of credits for each student. |  |  | | --- | |  | | |  | | --- | | - For each line, determine the tuition based on the district code: |  |  | | --- | |  | | |  | | --- | | - After processing all lines, display: |  |  | | --- | |  | |
|  | |  | | --- | | - If district code is "I", the cost per credit is 250.00. |  |  | | --- | |  | | |  | | --- | | - Total sum of all tuition owed. |  |  | | --- | |  | |
|  | |  | | --- | | - If district code is "O", the cost per credit is 500.00. |  |  | | --- | |  |      |  | | --- | | - Calculate the tuition owed for each student using the formula: tuition owed = credits \* cost per credit. |  |  | | --- | |  | | |  | | --- | | - Total number of students. |  |  | | --- | |  | |
|  | |  | | --- | | - Accumulate the total tuition owed and count the total number of students. |  |  | | --- | |  | |  |
|  |  |  |
|  |  |  |

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.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No user input | |  | | --- | | - Initialize first two Fibonacci numbers as 1, 1. |  |  | | --- | |  | | |  | | --- | | - Display the first 20 Fibonacci numbers. |  |  | | --- | |  | |
|  | |  | | --- | | - Use a for loop to compute the next Fibonacci number by adding the previous two numbers. |  |  | | --- | |  | |  |
|  | |  | | --- | | - Display each number in the Fibonacci sequence. |  |  | | --- | |  | |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |