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  int\_rate | principle = int(input(“User input”)  int\_rate = float(input(“User input”) |  |
| year = 1  int | For year in range (1,6, 1):  Display principle before calculations  Int = principle \* int\_rate  principle = principle + int  year = year + 1  tot\_int = tot\_int + int  display year count, principle after calculations  redo loop until 5 years is calucated  exit loop | Year (our loop iteration counter)  Principle at beginning of loop  Principle after calculations in loop |
| Tot\_int | #tot\_int = tot\_int + int (From our loop earlier)  Display tot\_int | The sum of the interest values from our 5 loop |

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 |
| numA  numB  count | numA = 0  numB = 1  count = 1 |  |
| numC | For count in range (1,21,1):  numC = numA + numB  print numC  numA = numB  numB = numC  count = count + 1 | numC will be displayed and the loop repeated. The new numC will be displayed for each iteration of the loop |
|  |  |  |
|  |  |  |

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 |
| Lastname  Salary | Open file  Get lastname from file  Get salary from file |  |
| bonus | While lastname !=””  If salary >= 100000  Bonus = salary \* 0.2  Elif salary >= 50000 and salary < 100000  Bonus = salary \*0.15  Else  Bonus = salary \*0.1  Display last name, salary and bonus | Prints last name, salary and bonus for each iteration of the loop |
| Bonussum | Bonussum = 0  Bonussum = bonussum + bonus | Sums the bonus continually for each iteration of the loop |
|  | Close the file  Print bonssum | Display the bonussum |
|  |  |  |

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.

Example Data File

Widget

10

50

Hammer

2

10

Saw

4

8

Etc

|  |  |  |
| --- | --- | --- |
| Input | Process | Output |
| Item | Count = 0  Totalextprice = 0 |  |
| Quantity | Open file  Get item  While item != “”  Get quantity and price from file  Extprice = quantity \* price  Count = count +1  Totalextprice = toalextprice + extprice  Display item, quantity, price and extended price  Get next item from file | Loops until the file has been read all the way through. Preforms calculations for the extended price based on data from the file.  Displays the relevant information from each loop iteration |
| Total extended price  Avgorder  count | Display count  Avgorder = totalextprice / count  Display avgorder  Display totalextprice | Calculates and displays the count # of loops, average order cost and the sum of all the extended price |
|  |  |  |

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.

|  |  |  |
| --- | --- | --- |
| Input | Process | Output |
| Lastname  district | Open file  Get lastname from file | This will read our first line of our file, which is the first student’s last name |
| Tuitition  numcredit | Sumoftuition = 0  Count = 0  While lastname !=””  Get district from file  Get numcredit from file  If district == “I”:  Credit = 250  Else  Credit = 500  Tuition = credit \* numcredit  Count = count +1  Display lastname, credits, tuition  Totaltuition = totaltuition + tuition  Get lastname from file | This will loop so long as there is a line in our file to be read  This will display the student, number of credits taken, and the tuition calculation for each loop iteration |
| totaltuition | Exit loop and close file after  Print count  Print total tuition | The count acts as the “number of students”  Our total tuition was summed in our loop, this displays that sum |
|  |  |  |

Example file

Jones

I

12

Adams

I

10

Baker

O

12

Smith

O

16