CIS 106 – Loops Part 2 Linar Jonasson, assignment logic 8

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 |
| principle | Get principle |  |
| rate | Year=0  Get rate  while year>5  YearInterest=principle\*rate  Print principle, principle+YearInterest  Principle=principle+YearInterest  Year=year+1 | Beginning end  Year 1: principle beginning+interest  Year 2: Year1End Year1End+interest  Year 3: Year2End Year2End+interest  Year 4: Year3End Year3End+interest  Year 5: Year4End Year4End+interest  Total interest earned: Year 5 end(or Year4End+interest) |

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 |
| No real input, it is premeditated | ##Wanted to do For loop until I found that While seemed better to use OR that I was unable to make use of it.  a=1  b=0  count=1  print(count,”: “,1)  while count < 21  b,a=a,a+b  print(count,”: “,a)  count=count+1 | Fibonacci sequence 1 to 20 and values ranging from 1 to 6765 |
|  |  |  |

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.

|  |  |  |
| --- | --- | --- |
| input | process | Output |
|  | Open file  TotalBonus=0 |  |
| Name+salary | Name=str(f.readline().rstrip(‘/n’))  While name !=””:  Salary=float(f.readline())  If salary>=100 000:  Rate=0.2  Elif salary>=50 000:  Rate=0.15  Else:  Rate=0.1  Bonus=salary\*rate  totalBonus=totalBonus+Bonus  print(name, salary, bonus, /n totalBonus)  Name=str(f.readline().rstrip(‘/n’)) | Name, salary, bonus  Total Bonus  #Repeat Until Text Is Wrong Or Ends |
|  | ##  I tried learning how to use IPOs online and from your IPO video, but I can’t seem find the rules for the usage, so I just assume that as long as the reader understands the logic, it’s fine. Unlike Flowgorithm where the logic has to be programmed in a flawed compiler. |  |

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 | output |
|  | Count=0  Total Extended Price=0 |  |
| Item(quantity, price enters immediately after, because it’s a text file, so no delayed input) | Read item  While item !=””  Read quantity, price  extendedPrice=quantity\*price  count=count+1  Total Extended Price = Total  Extended Price+extendedPrice  Print( item, quantity, price, extendedPrice)  Read next item | Item, Quantity, Price, Extended Price |
|  | Average=totalExtendedPrice/Count  Print Total Extended Price, Count, Average | Total Extended Price ,count, average |
|  |  |  |

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 |
|  | Students=0  totalTuition=0 |  |
| name | Read name  While name !=””  Read code, credits  If code==”I”:  tuitionCredit=250.00  else:  tuitionCredit=500.00  tuition=tuitionCredit\*credits  totalTuition=totalTuition+tuition  students=students+1  print name, credits, tuition  Read name | Name, credits, tuition |
|  | Print totalTuition, students | totalTuition, students |