Intro to Python – Lesson 33

Now that we created files with all this wonderful data, what are we going to do with it. I have included a very simple data file called Sales.dat. This includes an Invoice Number, Customer name, Item Cost, HST and Total Cost. Check out the following code that will read each line from the file, process it, and print the results. Do some research and find out what the split() and strip() functions do in this exercise.

```
f = open("Sales.dat", "r")
for SalesDataLine in f:
    SalesLine = SalesDataLine.split(",")
    EmpNum = SalesLine [0]
    EmpName = SalesLine [1]
    ItemCost = float(SalesLine[2])
    HST = float(SalesLine [3])
    TotalCost = float(SalesLine[4].strip())
    print(" {} {:<20} ${:,.2f} ${:,.2f} ".format(EmpNum, EmpName, ItemCost, HST, TotalCost))
f.close()</pre>
```

Add the data file to a new project and write a program that will read each line in the file, assign the values to variables, and print the values as seen from the code above.

Creating Reports

One of the most common reasons to process a file is to generate a report. A report basically takes the data from a file and produces a document that people can use for making decisions. Look at the following example. Reports are made up of heading lines – printed before the loop, detail lines – generated inside a loop as you process each line, and summary lines – normally counters and accumulators and is printed after the loop.

Use the table called Customer.dat to create the following reports. The file contains the Customer Account Number (Text), the Customer Name (Text), the Street Address (Text), the City (Text), the Phone Number (Text), the Balance Due (Float) and the Credit Limit (Float). A sample is below:

```
12345, John Doe, 123 Main St., St. John's, 754-4858, 382.00, 2000.00
12537, Sally Jones, 25 Airport Rd., St. John's, 579-1941, 2345.00, 2000.00
```

:

The first report is just a listing of customers and their contact info. Note the counter at the end of the report. This is a variable named CustCtr (or other similar name) that is set to 0 before the loop, incremented by one inside to loop as you process a customer, and printed after the loop.

The following report has a bit more detail and some calculations. This time there is a counter for the customers listed and accumulators for the balance due and minimum payment. The accumulator totals the values in each column. Variables such as BalDueAcc and MinPayAcc are again set to 0 outside the loop, updated based on the value being totalled, and printed outside the loop at the end. For extra fun, when you work with an accumulator, say it like a Klingon LOL.

123456789	1 2 0123456789012345678	3 4 90123456789012	5 234567890123	6 34567890123
WIDGITS INCORPORATED				
AC	COUNTS RECEIVABLE S	UMMARY REPORT	AS OF MM-DI	D-YYYY
ACCOUNT NUMBER	CUSTOMER NAME	BALANCE DUE	CREDIT REMAINING	MINIMUM PAYMENT
XXXXX	======================================	\$\$,###.##	\$\$, ###.##	\$\$,###.##
XXXXX	xxxxxxxxxxxxxxx	: : \$\$,###.##	\$\$,###.##	\$\$,###.##
Customer	s listed: ###	\$\$#,###.##		\$\$#,###.##
END OF LISTING				

Calculate the credit remaining as the balance due subtracted from the credit limit. The minimum payment is calculated as 10% of the balance due if the customer has not exceeded their credit limit, If the customer has exceeded their credit limit, the minimum payment will be 10% of the balance, plus the amount that the customer is over their credit limit.

See you at 1.