Intro to Python – Lesson 29

Today we want to look at an introduction to text files to store data. We saw that we can use lists to store multiple values, but since this is memory, it is destroyed when the program is over or the computer is turned off. We need a way to store data permanently. Check out the following video.

https://www.youtube.com/watch?v=0C2405R-uGk

Note that text files are the simplest for of storing data – as you move forward in your program, you will see how databases and tables are much more efficient. Take a bit of time and research the difference between text files (also referred to as Flat Files) and Database tables.

Use the following sample to write data to a data file. In most cases, data files of this type are set up as comma separated files – which means each value is separated by commas. Also note that all values pertaining to a single transaction are generally recorded on one line – for example one invoice per line or one customer per line. In most cases when using files you will open the file, process the file – in this case write to it, then close the file. Note that numeric values are converted to strings as they are written to the file.

Practice

The **NL Chocolate Company** needs a program to process salesperson travel claims when they return from a business trip. As employees return from business trips they record all required information on a Travel Claim Form, and return the form, with all invoices, to the main office.

The program will process the travel claims returned to the office. Initialize a constant for Claim number (34) and the HST rate (15%). Start by prompting the user to enter information from the Travel Claim Form including the employee number, name, location of the trip, start date, end date, the number of days (BONUS: rather than input the number of days use the start and end dates to calculate this value), a value to indicate if they used their own car, or if a car was rented (O or R), and the total kilometers traveled. Only enter the total kilometers if the employee used their own car. NOTE: No validations are required at this time.

Calculate the per diem amount by multiplying the total days by a daily rate of 85.00 for claims of 3 days or less, or by a daily rate of $100.00 for claims of 4 or more days. The mileage amount is calculated using a rate of .10 per kilometer if the salesperson used their car, or a rate of $56.00 per day is the salesperson rented a car. The Claim Amount is calculated as the Per Diem amount and the Mileage amount. The HST is calculated on the per diem amount only. The Claim Total is the Claim Amount plus the HST.

The program will display all input and calculated values to the screen as results. Only display the mileage amount if it is calculated. Just do a basic printout with headings and values.

*****Here is where the data file comes in – with a couple of other interesting concepts with the message and updating the constants*****

Write all the input values and the calculated values to a file called Claims.dat – separate each value in the row with a comma – this creates a comma separated file. Note that the only calculated value written to the file is the ClaimTotal. This is because any calculated value can be recalculated as needed since you have all the input values.

```
f = open("Claims.dat", "a")

f.write("{}, ".format(str(ClaimNum)))
f.write("{}, ".format(str(CurrDate)))   # This is the current system date
f.write("{}, ".format(EmpNum))
f.write("{}, ".format(EmpName))
f.write("{}, ".format(Location))
f.write("{}, ".format(StartDate)))
f.write("{}, ".format(EndDate))
f.write("{}, ".format(str(NumDays)))
 f.write("{}\n".format(str(ClaimTotal)))

f.close()
```

Display a message for the user indicating that the claim information has been saved and update the invoice number by adding 1 to it.

Repeat the program until the user enters the word END for the employee number.

Practice with the following. Notice how the lists are written to the file.

Write a program for a real estate agent to record information on home listings. Include the Listing number (9-digit number), street address, number of bedrooms, number of bathrooms, total square footage, the listing price, and the date – note that a home can have multiple prices – so store the date and price for each. Finally enter the status – must be one of Open, Offer Pending, or Sold. ****This is already completed in a previous lesson.**

Write all the input values to a file called Homes.dat. **BONUS: Set up a progress bar – or something similar – to show the user the date is saving.**

Display a message for the user indicating that the claim information has been saved.

Repeat the program until the user enters the word END for the listing number.

See you at 1.