Intro to Python – Lesson 31

In the example yesterday we set up a couple of constants at the beginning of the program. Most important is what happens to the Invoice number. Every time you start the program it will get reset back to the value in the code. We want this number to continue where it ended yesterday. Solution is to store the default values in a file.

In the same folder as the program, create a text file called Def.dat. Open the file and place the initial values for the Claim number and HST Rate with each value on a separate line (and any other defaults you want). The file will appear as follows:

```
34
.15
```

At the beginning of the program, add statements that will read the values from the file and assign them to variables. Note that all values in a file are strings so they can be converted to int or float. If you added other default / constant values, you will need to read them as well.

```
# Open the defaults file and read the values into variables
f = open('Def.dat', 'r')
ClaimNum = int(f.readline())
HSTRate = float(f.readline())
f.close()
```

In the main loop for the program, you can now use these values as required.

```
while True:
    # Input all required variables
        :
     # Do all the processing now
     HST = ItemCost * HSTRate
         :
     # Display the results
     print()
     print( … ClaimNum …)
     print()
         :
     # Write the values to a file for future reference.
     f = open(Stuff.dat”, “a”)
     f.write("{}, ".format(str(ClaimNum)))
     f.write("{}, ".format(CustName))
           :
     f.close()

    #Add 1 to the invoice number in preparation for the next one
     ClaimNum += 1
```

Finally, after the loop at the end of the program, write the values back to the defaults file.

```
# Write the current values back t the default file. Note the use of “w” to overwrite and the use of the \n so that each value is placed on a separate line.

f = open('Def.dat', 'w')
f.write("{}\n".format(str(ClaimNum)))
f.write("{}\n".format(str(HSTRate)))
f.close()
```

Now the next time you write the program it will read the values again at the top of the program, and the Invoice number will continue where you left off.

Give it a try.

Make the following changes to the file for the Movie Company we completed last class.

- Delete the file Movie.dat if it is on the system. Add an input after the movie name for the release date in the form YYYY-MM-DD.

- **Create a text file called Defaults.dat** which includes the Next Movie Number, the HST Rate, the Number of days for a New Release, and the number of weeks before the movie must be removed from the shelf. The file will appear as follows:

  ```
  10285
  .15
  45
  100
  ```

- **As the program starts**, read the values from the defaults file to set up your program constants – use some print() statements to check and make sure the values are correct. **At the end of the program**, after the end of the loop (This can even be done inside the loop after all other processing is complete – really keeps up-to-date), write the current values back to the file so they will be up to date the next time we run the program.

- **Add the following calculations** after the inputs and before the file update. Calculate the HST using the Rental cost and the rate from the defaults table. The Total Rental Cost is the Rental cost plus the HST. Determine the New Release end date based on the release date entered plus the days for a new release from the defaults table. Finally determine the Obsolescence date as the release date plus the number of weeks to be removed from the shelf. Display all calculated values with basic headings and formats.

- **After the information is added to the file**, display a message that reads “Movie information for XXXXXXXXXXXXXXX successfully saved. Press any key to continue.” The movie name is to be displayed in place of the X’s. **BONUS:** Wait 2 seconds before displaying this message but **let the user know something is happening during this time** – users like that.

See you at 1.