Intro to Python – Lesson 5

Please review the following video on "Formatting output" before the Python class.

https://www.youtube.com/watch?v=yVGSeRcQfyI

As opposed to a full program, we want to concentrate on the printed output. Set up a program that defines each of the variables, then write the print statements to generate the following results. For example – note that character values are enclosed in "" and numbers are set up as the bare numbers with no formatting. These values would have normally been input or calculated in the program.

Set up variables for each required value. These could have been inputs or calculations.

```
DateProcessed = "09-03-20"
SalesPerson = "Billy Joe McAllister"
Location = "Montreal"
StartDate = "08-15-20"
EndDate = "08-22-20"
NumDays = 7
DayCharge = 658.56
Mileage = 546
MilCharge = 132.54
HST = 102.84
ClaimTotal = 893.94
```

Print out the formatted results. X's represent string variables and #'s represent numbers.

```
          1         2         3         4
  12345678901234567890123456789012345678901234567890
1              ABC COMPANY
2        CLAIM CONFIRMATION RECEIPT
3
4    Date processed: XXXXXXXX
5    Salesperson:    XXXXXXXXXXXXXXXXXXXX
6    Location:       XXXXXXXXXXXXXXXXXXXX
7    Dates:          XXXXXXXX to XXXXXXXX
8    ------------------------------------
9    Days:     ###   Charge:    $#.###.##
10   Mileage: ####              $#,###.##
11   Tax (HST @ 15%):           $#,###.##
12                              ---------
13   Claim total:               $#,###.##
```

Variables and literals / headings can all be printed with an f-string. In general string values are left aligned (<) and numeric values are right aligned (>). Dollar values generally contain a floating $, two decimal positions, and a comma to separate thousands.

To print literals or headings just enclose the value in “”’s and include any spacing as required. For lines 1, 2, 8 and 12:

```
print(f"               ABC COMPANY")
print(f"       CLAIM CONFIRMATION RECEIPT")
print(f"   ----------------------------------")
print(f"                              ---------")
```

For variables you include a {} to indicate where the variable is to be placed. For lines 4, 5 , 6, and 7 we are printing string variables. Using the f-string both the variable and the alignment are defined in the {}. Note that the s at the end of the alignment indicates a string.

```
print(f"   Date processed: {DateProcessed:>8s}")
print(f"   Salesperson:    {SalesPerson:<20s}")
print(f"   Location:       {Location:<20s}")
print(f"   Dates:          {StartDate:<8s} to {EndDate:<8s}")
```

Integer values are like strings in that the variable is specified by the {}. Generally, integers will be right aligned as on lines 9 and 10. Note the d at the end of the alignment indicate an integer.

```
print(f"   Days:     {NumDays:>3d}")
print(f"   Mileage: {Mileage:>4d}")
```

For dollar values we need to format the variable to currency and then print the value right aligned. Note the DayCharge and MilCharge on lines 9 and 10. Note the f at the end of the format statement indicates a float.

```
DayChargeDsp = "${:,.2f}".format(DayCharge)
MilChargeDsp = "${:,.2f}".format(MilCharge)

print(f"   Charge:     {DayChargeDsp:>9s}")
print(f"               {MilChargeDsp:>9s}")
```

It is possible with f-string to place the entire format and alignment in the {} as shown here. Basically you nest a second f-string that formats and aligns at the same time.

```
print(f"   Charge:    {f'${DayCharge:,.2f}':>10}")
print(f"              {f'${MilCharge:,.2f}':>10}")
```

Note on lines 9 and 10 that both the integer and dollar values are on the same line. Note that using the format() you need to set up the new variables but the f-string does not.

```
DayChargeDsp = "${:,.2f}".format(DayCharge)
MilChargeDsp = "${:,.2f}".format(MilCharge)

print(f"   Days:     {NumDays:>3d}   Charge:    {f'${DayCharge:,.2f}':>10s}")
print(f"   Mileage:  {Mileage:>4d}              {f'${MilCharge:,.2f}':>10s}")
```

Here is an exercise to practice. Note that line 8 contains the City, Province and Postal Code. Set up variables and assign appropriate values like we did in the first example.

```
          1         2         3         4         5         6         7
  12345678901234567890123456789012345678901234567890123456789012345678901234567890
1                        HONEST PETER'S GARAGE
2                          123 Fixit Street
3                       St. John's, NL  A1A 1A1
4   Invoice#: ####                                        Date: XXXXXXXX
5   ------------------------------------------------------------------------
6    Customer: XXXXXXXXXXXXXXXXXXXXXXXXXXXXX     Plate Number: XXXXXX
7    Address:  XXXXXXXXXXXXXXXXXXXXXXXXXXXXX     Mileage:      ######
8              XXXXXXXXXXXXXXXXX, XX  XXXXXX
9
10                                            Cost of Labor:  $#,###.##
11                                            Cost of Parts:  $#,###.##
12                                           Total Discount:  $#,###.##
13                                                     HST:  $#,###.##
14                                                           ---------
15                                            Invoice Total:  $#,###.##
16  ------------------------------------------------------------------------
17      Honest Peter's - There to meet the needs of our customer!!
18  ------------------------------------------------------------------------
```

See you at 1.