ITEC 1150 Chapter 1.2 Lab Projects

FORMATTING, INPUT & MORE ON DATA TYPES

Program Development Plan (PDP)

This is a step-by-step process for successfully constructing an application. Follow these steps and repeat until you have successfully completed the program.

This is a reminder to use the PDP process.

You do not need to turn in your PDP document, but going through the process will help you design your programs.

PDP template -

1. Problem definition
2. Analysis of variables & functions
3. Steps in algorithm
4. Code (separate .py file with added comments)
5. Screen snips from testing
6. Keep track of any wanted or needed improvements for a future version

General Requirements

All assignments must meet the following requirements:

The program must start with header at top and include appropriate comments throughout. Header example:

```
"""

Author: Erik Granse

Date: 2024-09-02

Description: Calculate and display student's average grades

"""
```

- Ensure the output is *information*; it needs to be a statement which explains the value being displayed (for example, “The average grade is 12.34”). Simply outputting “12.34” is not sufficient.

General Requirements (cont.)

All assignments must meet the following requirements:

- The data in the program must be stored in variables.
- The output **must** come from variables in the program
  - Do not simply hard code the output value in the `print()` statement.
  - Some data will be given to you, and some will be user input—any calculations that need to happen must be in your program. Don't calculate somewhere else and enter the value into your program.

Lab Section 1: Gas Calculator

- Create a program named `gas_calc.py`. The program must:
- Ask the user for
  - the number of miles they drove on a trip
  - the number of gallons of gas they used
  - the price of gas per gallon
- Use the input to calculate and display the cost of the trip & the miles per gallon (MPG) for the trip in table format.

- Your program must use the same alignment and column layout as the sample below (although the exact width does not need to match); note the placement of the dollar sign!

```
Here are some fun facts about your trip:
MPG               50.00
Trip cost      $   5.00
```

When output is shown, it's just an example – your program must work with different inputs!

Lab Section 2: Coffee Sales

- Create a program named `coffee_sales.py`.
- We started this program during the lesson. Now, assume that your coffee shop sells coffee, tea, and cappuccino. NOTE: part of your lesson code can be used as a model for the parts you need to add!
- Expand the program by asking the user for the number of cups of <u>each drink</u> sold, and how much each drink costs (6 separate questions).
- Display the total sales for each drink, and the total sales for everything.
- Print the information into a table using python 3 formatting (adapt the code from bus_fare2.py, done in class).

- Table requirements:
  - Column labels centered
  - String fields left-aligned
  - Integer fields left-aligned
  - Float fields right-aligned, and decimal points must align vertically
  - Dollar signs must be vertically aligned
  - Be aware that you you may have more or digits than you expect based on user input!

```
Drink Type   Cups Sold     Price       Total
Coffee           100     $  2.00    $ 200.00
Tea               10     $  1.55    $  15.50
Cappuccino         0     $  5.00    $   0.00
Total            110                $ 215.50
```

Lab Section 3: Area Calculation

- Create a program named `area.py`. The program must:
- Ask the user:
    - The measurement unit (inches, feet, centimeters, etc.)
    - The length of the rectangle in the units specified (e.g., "What is the length in inches?") Float values must be accepted.
    - The width of the rectangle in the units specified (e.g., "What is the length in centimeters?"). Float values must be accepted.
- Calculate the area of the rectangle (defined as length * width).
- Display the area as a full statement rounded to two decimal places (e.g., "The area of the rectangle is 12.70 square centimeters.")

```
Welcome to the Rectangle Area Calculator!
What is your measurement unit (in., ft., cm., etc.)? meters
What is the length of the rectangle in meters? 2.5
What is the width of the rectangle in meters? 3.6
Your rectangle is 9.00 square meters.
```

- NOTE: the values in italics at the end of lines 2-4 in the example are NOT hard coded. They are user input & could be different. Likewise, in the 3 print statements, the word "meters" and the number 9.00 are NOT hard coded. They are filled in from variable names & will change depending on user input.

Lab Section 4: Sales Tax

- Create a program named `sales_tax.py`: The program must:
- Ask the user to enter the price of a purchase order (float values must be accepted).
- Calculate state sales tax at 5%.
- Calculate county sales tax at 2.5%.
- Output a sales report showing Purchase Order Price, State Tax, County Tax, and Grand Total.
- Challenge: to avoid the appearance of math errors, it's best to use the `round()` function on the tax calculations – round each to 2 decimal places.
- When printing, use the `.format()` and/or f-string shortcut method to create a table as shown to the right. Test with sample data <u>and other values</u> to ensure it's a dynamic solution.

- Note the layout of the table below; your table should be very similar.

```
What is the total price of your purchase order? 753.25

Custom Delivery Sales Receipt
PO Amount      $    753.25
State Tax      $     37.66
County Tax     $     18.83
Total          $    809.74
```

Lab Section 5: Math Functions

- Create a program named `math_functions.py` :

  Paste these comments into your python file, as a heading for each task. Then write a line of code below each comment, to provide the result as shown.

```
# cut off 7.89 to 7

# round 54.345395 to 54.345

# calculate the square root of 2

# calculate the sin of 7

# display the value of pi

# display pi rounded to 3 decimal places
```

- **You may need to use online help for this.**
- Important: Add this import statement immediately after your file header:

```
import math
```

- You can then use functions with math to get the values you need to format, like so:

1. `math.sin(7) # the sin of 7`
2. `math.sqrt(2) # square root of 2`
3. `math.pi() # Pi, obviously`

Upload!

- In all files, don't forget your comments ("""docstring""" and # short), naming conventions and python 3 formatting style
- Lab projects to upload to D2L:
  - gas_calc.py
  - coffee_sales.py
  - area.py
  - sales_tax.py
  - math_functions.py

Questions? Let me know – before the deadline!