Brian Pochert
03/11/2025
IT FDN 110A (WIN2025)
Assignment 07
https://github.com/bpoch/IntroToProg-Python-Mod07

Module 07 Classes and Objects

Introduction

Module 07 expands our knowledge of Classes which were introduced to us in Module 06. We learned about additional ways to organize larger code. We can comprehend that large programs can have many statements, functions and classes. We dive deeper into Classes and the functions they contain. We learn how to create specific classes that contain data for our programs. The two major types of classes we learned about are Data Classes and Processing Classes. We also learned about attributes, constructors, and properties in how they relate to classes

Data Classes

Classes in Python center around specific uses in the program. One type of class is the Data class. While data classes can have functions, they usually have attributes constructors and properties. Data Class attributes are variables that store data inside a class. Properties are used to control access to the attribute. Constructors initialize an object when its created. We learned about getters and setters along with the python keyword "self" and the syntax (__init__). The concepts were robust and somewhat hard to follow. One can learn more about the Data Class in Python here:

https://www.datacamp.com/tutorial/python-data-classes (external site)

Processing Classes

A processing class in Python is a class that is used to perform operations on data opposed to storing it like a data class. These classes typically contain methods or functions for manipulating, transforming, or analyzing data. Processing Classes are often used for data validation, computation, and automating workflows.

If you want to learn more about Python Processing Classes see this article:

https://www.w3schools.com/python/python_classes.asp (external site)

Self and Inheritance

Module 07 showed us two other important concepts in Python programing. The "self" keyword and the idea of inheritance.

The "self" keyword is important as it is a reference to the current instance in the class. It is used to access attributes and methods (functions) in the class. When initializing a class, the first parameter should be self. The syntax is very confusing, but I was able to follow examples and get the code in Assignment 07 to work.

Inheritance was another concept we learned about. This concept was easier to digest. It basically means that you can allow a class to "inherit" the attributes and methods of another class. Another way to say this is passing class information to another class. The syntax tells Python which class is inherited. To inherit a class, you place the class name in parenthesis of another class like so:

```
Class Animal:

Class Dog(Animal):
```

In the example above, we created the class named "Animal". If we want to pass the information from Animal into our class named "Dog" we place "Animal" in parenthesis when we create the "Dog" class. This creates a parent/child hierarchy of the classes where Animal is the parent and Dog is the child.

Assignment 07

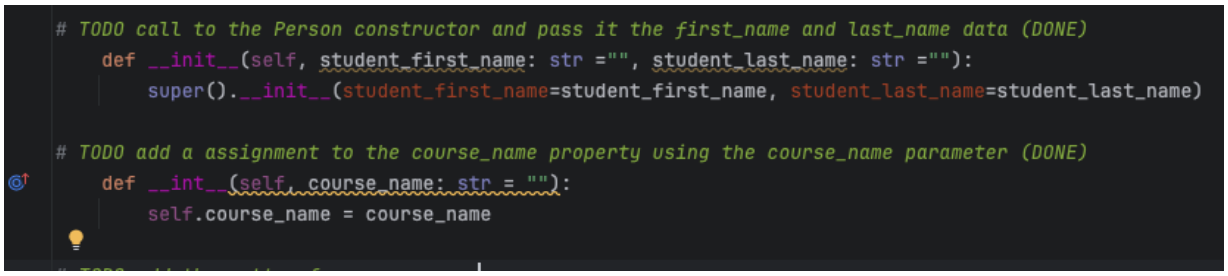
In assignment 07, we continue to modify and build on our student registration program. In this assignment we add a set of data classes called "Person" and "Student". The assignment included a starter program .py file. The starter program included 11 "TODO" comments which provided us with a blueprint on how to modify the program. The assignment also directed us to use Lab07-03 as a guide. Lab07-03 included a starer program and TODO list along with instructions on the exact code syntax needed to complete that lab's program.

I started my assignment by going through each TODO one step at a time and completing the required code. I used the Lab07-03 as a guide on how to create the lines of code. I found taking my time and going through each line of the code helped. It was easy to start seeing big picture of how the code is divided up into the classes and the specific functions in each class.

I ran into one problem in writing the code for the TODO:

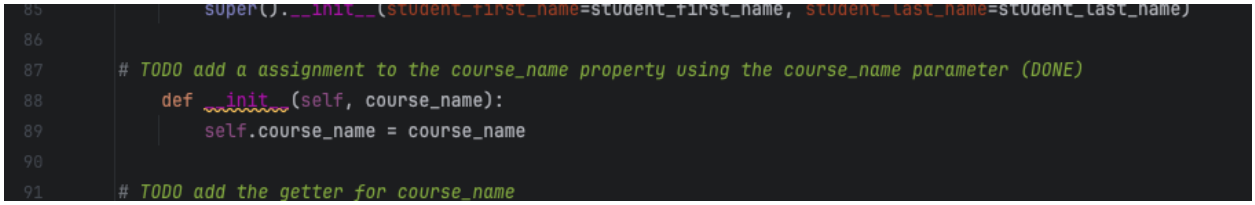
```
TODO add a assignment to the course_name property using the course_name parameter (Done)
```

I initially wrote the code this way in figure 1



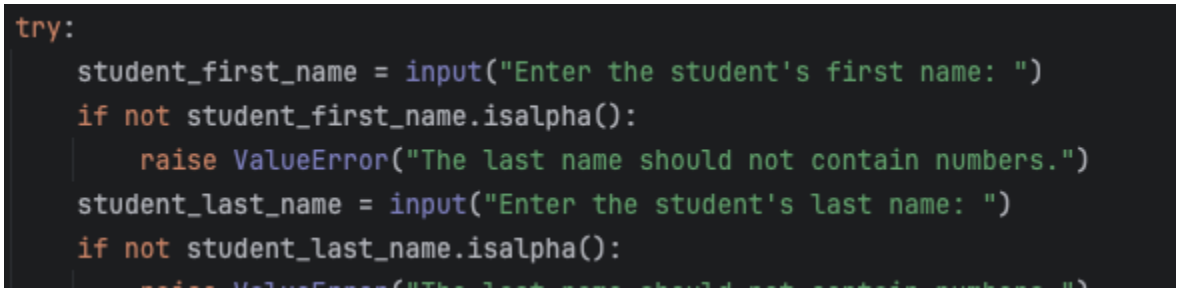
(figure 1)

This caused an issue with as seen with the indicator to the left of the code. I used ChatGPT and asked "how do I add an assignment to a property using a parameter in python". This returned several options which displayed the correct syntax for this TODO and the next two (TODOs). The following two (TODOs) asked for the "getter" and "setter" for the course_name variable. I was able to fix the code to this in figure 2



(figure 2)

I continued to review the program line for line and checked for errors. I found this error in the starter code shown in figure 3



(figure 3)

The custom error message for an issue with the first name stated the "last name should not contain numbers". I corrected this in the code and made a note in the change log. I also had to correct several of my indentations in the classes and methods. It did not seem to affect the program working or not, but I wanted it to look correctly formatted.

Summary

The assignment for Module 07 was conceptually very difficult. It was easier to complete because of the TODO lists and the example code in Lab07-03. So, while I was able to complete the code and the program works, I found it harder to understand exactly what I completed. I understand concepts of classes, constructors, and inheritance. I struggled with specific keywords and syntax to make everything work.