Adam Eichholz
August 27, 2025
Foundations of Programming: Python
Assignment 7
https://github.com/eichhoa1/IntroToProg-Python-Mod07

Classes and Objects

Introduction

In this paper, I look at the changes I made to the student enrollment project to incorporate inherited code. This week felt more about the philosophy behind certain design decisions, and as such, there were not a lot of changes ultimately made to our code. As a quick note, I am continuing to list my screencaptures as Figures in their own section, located after the Summary. This change has been made to try and reduce repeated spacing in the Topic body.

Topic

As I stated in the introduction, this week's assignment felt more about helping us to understand certain design theory when structuring code. I closely followed along with the notes and labs for this week in anticipation of the assignment, and while I understood what to do when writing my code, I cannot say that I came away with the best understanding of what the code actually means or why things are the way they are. I recognize that this is something that will develop with practice, however.

When actually starting the assignment, I was grateful that there were To-Dos clearly marked out for us this week. That was extremely helpful in understanding what changes I was expected to make where they would need to be. By following along from what I had written for Lab 3, things went smoothly and my code ran as expected.

Initially, I had attempted to write my code as "student_first_name" (e.g.), rather than simply "first_name" as in the lab, in an attempt to match what I remembered as the variable of the same name when we first wrote our program. However, I could not help but feel like I was over complicating my code, and decided to just write it as "first_name" instead. It might make sense to title those lines more accurately to match with the person, and if I had felt more confident, I would have used "student_first_name" instead.

Summary

While this week primarily focused on theory, I still developed a more practical understanding of how inherited code can work. I hope to continue to learn it through more practice and application.

Figures

Figure 1: Person Class

```python
# Processing ------------------------------------- #
class Person:
    """
    A class representing person data.

    Properties:
        first_name (str): The student's first name.
        last_name (str): The student's last name.

    Changelog:
        - Adam Eichholz, 8/26/2025: Created the class.
    """

    def __init__(self, first_name: str = "", last_name: str = ""):
        self.first_name = first_name
        self.last_name = last_name

    @property
    def first_name(self):
        return self.__first_name.title()

    @first_name.setter
    def first_name(self, value: str):
        if value.isalpha() or value == "":
            self.__first_name = value
        else:
            raise ValueError("The student's first name should not contain numbers.")
```

Figure 2: Student (Person) Class:

```python
class Student(Person):
    """
    A class representing student data.

    Properties:
        first_name (str): The student's first name.
        last_name (str): The student's last name.
        course_name (str): The name of the course the student is enrolling into.

    Changelog:
        Adam Eichholz, 8/26/2025: Created the class.
        Adam Eichholz, 8/26/2025: Added properties and private attributes.
        Adam Eichholz, 8/26/2025: Moved first_name and last_name into a parent class.
    """

    def __init__(self, first_name:  str = '', last_name: str = '', course_name: str = ''):
        super().__init__(first_name=first_name, last_name=last_name)
        self.course_name = course_name

    @property
    def course_name(self):
        return self.__course_name
```