James Copsey

11/20/2024

ITFDN110

Assignment06

https://github.com/jcopsey/IntroToProg-Python-Mod06/

Assignment 06 – Functions

Introduction

Assignment 06 is focused on teaching us how classes and functions can be utilized to write code that is easily maintained and understandable at a glance. This is a design principle of software development called the Separation of Concerns (SoC).

We are creating a program to the program we wrote in Assignment05, however, this time we're developing functions to achieve the same goal.

Importing Functions

I took a risk for Assignment06 and did something that wasn't explicitly asked for in the assignment document.

Learning about Separation of Concerns got me thinking about how we could take that a step further. We're declaring all these classes and functions in a file and then calling them all in that same file. What if we pulled those classes out into a new file and imported them into the 'main' file where we call them.

That's exactly what I did with the file titled 'functions.py.' I've included it along with 'Assignment06.py' and 'Enrollments.json.' As long as you're running the code from the same directory, the import at the top of 'Assignment06.py' will allow the classes from 'functions.py' to be used.

Here is the import line at the top of 'Assignment06.py':

```
from functions import *
```

Ideally, I wouldn't be importing every class from the 'functions.py' file, however, we are using every function in 'Assignment06.py' except for output_error_messages().

Python Enhancement Proposal 8

According to Python Enhancement Proposal (PEP) 8, a line of code should be no more than 79 characters.

My code can appear disjointed at times but I was doing the best I knew how to stay within those 79 characters and still have a working program.

Here's one of the functions I made where you can see an example of that:

```
    @staticmethod
    def input_student_data(student_data:list):
        """
        Gets first + last + course name from user and
        appends current list of students and registered courses

        ChangeLog:
        jcopsey,11/20/2024,Created function

        :return: Updated students list
        """
        students: list = student_data
        try:
            student_first_name : str = input(
                "Enter the student's first name: ")
            if not student_first_name.isalpha():
                raise ValueError("The last name should not contain numbers.")
            student_last_name : str = input("Enter the student's last name: ")
            if not student_last_name.isalpha():
                raise ValueError("The last name should not contain numbers.")
            course_name = input("Please enter the name of the course: ")
            student_data = {"FirstName": student_first_name,
                            "LastName": student_last_name,
                            "CourseName": course_name}
            students.append(student_data)
            print((
                f"\nYou have registered {student_first_name} "
                f"{student_last_name} for {course_name}."
            ))
        except ValueError as e:
            IO.output_error_messages(message="Invalid value.", error=e)
        except Exception as e:
            IO.output_error_messages(e.__str__())
        finally:
            return students
```

Summary

We learned about classes and functions and how to use them to create a separation of concerns so that our code is readable and maintainable. Pulling classes out into a separate file and then importing them as needed is a way to declutter a program.

Our program achieves almost exactly the same result as the program we wrote in Assignment05, but in a way that allows for a lot of flexibility moving forward.