File: Assignment06_Description_CharlesLloyd.pdf
GitHub Repository Link: https://github.com/ChasLloyd/IntroToProg-Python-Mod06.git

Assignment 6 for

Foundations of Python Programming

Charles J Lloyd, Student, University of Washington

11/18/2025

Introduction

This document describes the development and testing of a Python script demonstrating the use of classes, functions (methods), and the "separation of concerns" pattern for organizing a program.

This assignment was completed using the PyCharm IDE (version 2025.2.3) and Python Version 3.14.0 (64-bit). The GitHub platform was used to distribute this script and related documents.

This work was completed in fulfillment of the requirements for one of eight homework assignments for the course IT FDN 110 A Au 25: Foundations of Python Programming. This course was offered by the University of Washington (UW) department of Professional & Continuing Education. The instructors for this asynchronous on-line course include:

- Randal Root, Instructor
- Becky Peltz, Instructional Assistant

All files associated with this course were developed and tested on a Dell Inspiron 16 Plus 7620 laptop computer running Microsoft Windows 11 Home, Version 10.0.26100 Build 26100. The files were stored locally in the directory C:/Python/PythonCourse.

Procedure

Prior to working on the coding portion of the assignment, the student read the notes and viewed the videos indicated in the documents "Mod06-Notes.docx" and "Mod06-Assignment.docx", provided for this course on the UW Canvas site. Section 4 of this

document contains the “Acceptance Criteria” for this assignment which essentially make up the “Project Specification.” The steps involved in creating this program included:

- Reviewed the project specification
- Copied and updated the standard header created for previous assignments
- Added the classes, methods, constants and variables required by the specification
- Tested each method independently using custom code
- Updated the body to run methods for each menu choice
- Tested code for many combinations of menu choice, with and without the Enrollments file in the default directory
- Updated the external documentation (this document)
- Uploaded all files to the GitHub repository

Program Organization

Following a “separation of concerns” pattern, the program is organized into the following sections:

- HEADER
- DIRECTIVES
- DATA LAYER
    - Constants
    - Variables
- PROCESSING LAYER
    - Class FileProcessor
- PRESENTATION LAYER
    - Class IO
- BODY

Upon startup the program opens and reads the data file using a method from the FileProcessor class named “read_data_from_filt.” File-related error conditions are managed using a try-except structure that calls the method “output_error_messages” in the IO class to format and display the messages.

From that point on control is managed using an infinite while loop that presents the menu and waits for the user to select a menu option. Menu presentation and the selection of an option are accomplished using two methods from the IO class: “output_menu” and

"input_menu_choice." The user can stop the program by breaking out of the while loop by selecting menu option 4.

Menu Option 1: Register a Student for a Course

When this option is selected the program calls the method "input_student_data" contained in the IO class.

This method queries the user to enter the first and last names and the name of the course. Structured error handling is accomplished using try-expect statements and checking to assure the first and last names contain only alphabetic characters. If any errors occur within the try section, both custom and Python-generated messages are presented using the IO.output_error_messages method after which the program returns to the menu.

Menu Option 2: Show current data

When this option is selected the program calls the method "output_student_courses" in the IO class. This method prints out the student records in one of three styles:

print_style = 1: Includes keys, quotes, and brackets
print_style = 2: Includes explanatory text
print_style = 3: Comma separated values, no keys, quotes or brackets

Print style 3 conforms to the requirements for this project.

Menu Option 3: Save data to a file

When this option is selected the program calls the "write_data_to_file" method contained in the FileProcessor class.

When selected, this method reformats the data, received as a list of dictionaries, to conform to the JSON file format and dumps student data to the "Enrollments.json" file. A try-except structure and structured error handling are used within this method.

Menu Option 4: Exit the program

Upon selection of this option the program breaks out of the match-case structure that manages the menu selections after presenting the message "Exited the program"

Invalid Menu Selections

The program handles the case of an invalid menu selection inputs (any keys other than 1 to 4) using an else statement that returns program control to the menu (match statement) after presenting the message "Invalid Menu Choice."

Test Results

The Python script in the file “Assignment06.py” was tested in three ways using: the PyCharm IDE, a Windows terminal, and Notepad. Testing was completed both with and without the Enrollments.json file located in the default directory. In this testing the following behaviors were verified:

- The program takes the user's input for a student's first, last name, and course name.
- The program displays the user's input for a student's first, last name, and course name.
- The program saves the user's input for a student's first, last name, and course name to a JSON file.
- The program allows users to enter multiple registrations (first name, last name, course name).
- The program allows users to display multiple registrations (first name, last name, course name).
- The program allows users to save multiple registrations to a file (first name, last name, course name).

The program ran correctly for all combinations of test conditions. Testing from a Windows terminal was accomplished using the commands:

```
cd c:\Python\PythonCourse\_Module06\Assignment

python Assignment06.py
```

Distribution

The following three files were uploaded to the GitHub repository “IntroToProg-Python-Mod06”:

- Assignment06.py
- Assignment06_Description_CharlesLloyd.pdf
- Enrollments.json

The link for this repository is provided at the top of this document. These three files were also uploaded to the course directory on the Canvas website.