PROFESSIONAL & CONTINUING EDUCATION
UNIVERSITY of WASHINGTON

Name: Joseph Rideout
Date : November 22, 2023
Course: Using functions
Mod06-Assignment 06

Python Functions

Introduction

This document introduces the sixth assignment, which showcases the utilization of class, functions, and parameters. Covering the fundamentals, including parameters and arguments.
Additionally, I found Mr. Randel Root live sessions to be helpful for understanding python for this week assignment.

Class

In Python, A class is a blueprint or a template for creating objects. A class defines the properties and behaviors that objects created from it will possess. Once a class is defined, you can create multiple instances of that class with their own unique data but sharing the same structure and behavior defined in the class.

Functions

A function is a block of reusable code that performs a specific task or a set of instructions. Functions are defined using the **def** keyword followed by a function name, parentheses **()**, and a colon ':' to indicate the beginning of the function block. Once defined, functions can be called by their names, and arguments can be passed into them.

Parameters and Arguments

Parameters are the variables listed in the function definition. They represent the data that a function expects to receive when it's called. Parameters are used within the function's definition to perform operations.
Arguments are the actual values passed to a function when it's called. They fulfill the parameters' expectations. They can be constants, variables, expressions, or any other Python object.

Assignment 06

For this week's assignment, the task is to enhance a menu-driven program by implementing code that accomplishes the following tasks: defining a class structure, improving error handling, adding additional functionalities, and enhancing the overall user interface.
Here is an Example of Python Code: (Figure 1)

```
# ------------------------------------------------------------------------------------------ #
# Title: Assignment06_Starter
# Desc: This assignment demonstrates using functions
# with structured error handling
# Change Log: (Who, When, What)
#   JRideout,11/21/2023,Created Script
#   <Joseph Rideout>,<11/21/2030>,<Created Script>
# ------------------------------------------------------------------------------------------ #
import _io
import json

# Define the Data Constants
MENU: str = '''
---- Course Registration Program ----
  Select from the following menu:
   1. Register a Student for a Course.
    2. Show current data.
   3. Save data to a file.
    4. Exit the program.
-----------------------------------------
'''
# Define the Data Constants
# FILE_NAME: str = "Enrollments.csv"
FILE_NAME: str = "Enrollments.json"

# Define the Data Variables and constants
# Variables
menu_choice = ""
students = []



# Classes
class FileProcessor:
    """Handles file-related operations."""


    # When the program starts, read the file data into a list of lists (table)
    # Extract the data from the file
    @staticmethod
    def read_data_from_file(file_name: str, student_data: list):
        """Reads data from a file and populates the student data."""
        try:
            with open(file_name, "r") as file:
                student_data.extend(json.load(file))
                IO.output_student_courses(student_data)
        except FileNotFoundError as e:
            IO.output_error_messages("Error: File not found.", e)
        except json.JSONDecodeError as e:
            IO.output_error_messages("Error: JSON decoding issue.", e)
        except Exception as e:
```

```
            IO.output_error_messages("Error: Issue reading file.", e)

    @staticmethod
    def write_data_to_file(file_name: str, student_data: list):
        """Writes data to a file."""
        try:
            with open(file_name, "w") as file:
                json.dump(student_data, file)
            IO.output_student_courses(student_data)
        except Exception as e:
            IO.output_error_messages("Error: Issue writing to file.", e)


class IO:
    """Handles input/output operations."""

    @staticmethod
    def output_error_messages(message: str, error: Exception = None):
        """Outputs error messages."""
        print(message)
        if error:
            print("-- Technical Error Message -- ")
            print(error.__doc__)
            print(error)

    @staticmethod
    def output_menu(menu: str):
        """Outputs the menu."""
        print(menu)

    @staticmethod
    def input_menu_choice():
        """Takes user input for menu choice."""
        return input("What would you like to do: ")

    @staticmethod
    def output_student_courses(student_data: list):
        """Outputs student course data."""
        print("Data stored in file:")
        for student in student_data:
            print(f"Student {student['FirstName']} {student['LastName']} is enrolled in
{student['CourseName']}")

    # Input user data
    @staticmethod
    def input_student_data(student_data: list):
        """Takes input for student data."""
        try:
            student_first_name = input("Enter the student's first name: ")
            if not student_first_name.isalpha():
                raise ValueError("The first name should only contain letters.")
            student_last_name = input("Enter the student's last name: ")
            if not student_last_name.isalpha():
                raise ValueError("The last name should only contain letters.")
            course_name = input("Please enter the name of the course: ")
            student_data.append(
                {"FirstName": student_first_name, "LastName": student_last_name, "CourseName":
course_name})
```


```
        except ValueError as e:
            IO.output_error_messages("Error: Invalid name entered.", e)
        except Exception as e:
            IO.output_error_messages("Error: Issue with input.", e)



# Main program logic
if __name__ == "__main__":
    FileProcessor.read_data_from_file(FILE_NAME, students)

    while menu_choice != "4":
        IO.output_menu(MENU)
        menu_choice = IO.input_menu_choice()

        if menu_choice == "1":
            IO.input_student_data(students)
        elif menu_choice == "2":
            IO.output_student_courses(students)
        elif menu_choice == "3":
            FileProcessor.write_data_to_file(FILE_NAME, students)
        elif menu_choice != "4":
            print("Please only choose option 1, 2, 3, or 4")

    print("Program Ended")
```

Results: PyCharm / Command Prompt

PyCharm

```
Data stored in file:
Student Vic Vu is enrolled in Python100

---- Course Registration Program ----
  Select from the following menu:
    1. Register a Student for a Course.
    2. Show current data.
    3. Save data to a file.
    4. Exit the program.
----------------------------------------

What would you like to do: 1
Enter the student's first name: Joe
Enter the student's last name: Rideout
Please enter the name of the course: Python100
```

```
---- Course Registration Program ----
  Select from the following menu:
    1. Register a Student for a Course.
    2. Show current data.
    3. Save data to a file.
    4. Exit the program.
----------------------------------------

What would you like to do: 2
Data stored in file:
Student Vic Vu is enrolled in Python100
Student Joe Rideout is enrolled in Python100
```

```
What would you like to do: 3
Data stored in file:
Student Vic Vu is enrolled in Python100
Student Joe Rideout is enrolled in Python100

---- Course Registration Program ----
  Select from the following menu:
    1. Register a Student for a Course.
    2. Show current data.
    3. Save data to a file.
    4. Exit the program.
----------------------------------------
```

```
---- Course Registration Program ----
  Select from the following menu:
    1. Register a Student for a Course.
    2. Show current data.
    3. Save data to a file.
    4. Exit the program.
----------------------------------------

What would you like to do: 4
Program Ended

Process finished with exit code 0
```

Summary

This document demonstrates the basic functions and methods used in Python programming.
The knowledge gained from this week's assignment was an essential part of understanding Python 3 programming.