NAMAL UNIVERSITY MIANWALI

DEPARTMENT OF ELECTRICAL ENGINEERING

DATA STRUCTURE AND ALGORITHM

LAB # 03

REPORT

Title : Introduction to OOP

Name	*Fahim-Ur-Rehman Shah*
Roll No	*NIM-BSEE-2021-24*
Intructor	*Ms. Naureen Shaukat*
Lab Engineer	*Mr .Ali Hasnain*
Date	*06-April-2023*
Marks	

Introduction :

In previous lab, we have studied the loops and functions, with and without index.

Lab Objectives :

The objective of this lab is to introduce students to the concept that Python is an object-oriented programming language. In this lab students will enable the concepts of Defining a class, constructing object, accessing members of objects, and selfparameter. Another objective of this lab is to familiarize students to the concept of files in python.

Lab Outcomes :

- CLO:1 The students will enable the concepts of defining a class.
- CLO:1 The students will be able to construct object.
- CLO:1 The students will learn how to access members of objects.
- CLO:1 The students will learn the concept of self-parameter.
- CLO:1 The students will learn about Files in python.
- CLO: 3 A report on the basis of lab tasks and home tasks.

LAB TASKS

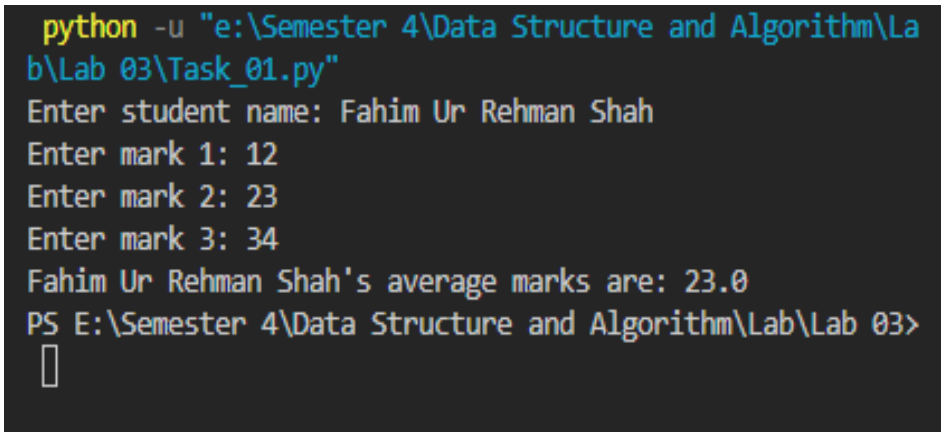
Task 1 : Write a Python program that creates a Student class with an __init__ method that takes a name and marks argument and initializes corresponding instance variables.
The average method should calculate and return the average of the marks list.
The program should prompt the user to enter the name of a student and their marks for three subjects.
It should then create a Student object with the given name and marks, and print the student's name and their average marks.

CODE :

```
class Student:                              # Creating a class name student
    def __init__(self, name, marks):
        self.name = name
        self.marks = marks

    def average(self):
        return sum(self.marks) / len(self.marks)

name = input("Enter student name: ")
```

```
marks = []
for i in range(3):
    marks.append(int(input(f"Enter mark {i+1}: ")))

student = Student(name, marks)
print(f"{student.name}'s average marks are: {student.average()}")
```

Output Screen Shot:



Explanation :

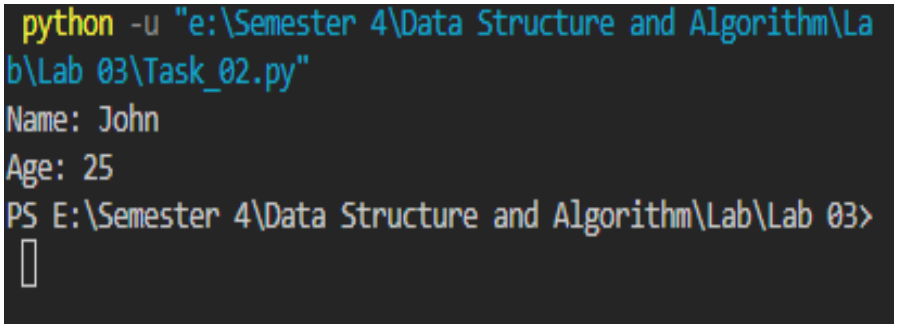
- We define a class named Student with an **init** method that takes a name and marks argument and initializes corresponding instance variables.
- The average method calculates and returns the average of the marks list using the sum and **len** functions.
- We prompt the user to enter the name of a student and their marks for three subjects using the input function and a for loop that iterates three times.
- We create a Student object with the given name and marks using the Student class and the input values.
- We print the student's name and their average marks using the print function and formatted strings with the student object's attributes and methods.

Task 2 : Write a class called Person which has a constructor to assign initial values to name and age. Then call a method to display person's name and age.

CODE :

```
class Person:
    def __init__(self, name, age):
        self.name = name
        self.age = age

    def display_info(self):
        print(f"Name: {self.name}\nAge: {self.age}")

person = Person("John", 25)
person.display_info()
```

Output Screen Shot:



Explanation :

- We define a class named Person with a constructor that takes two arguments, name and age, and initializes corresponding instance variables.
- We define a method named **display_info** that prints the person's name and age using the print function and formatted strings with the person object's attributes.
- We create a Person object with the name **"John"** and age **25** using the Person class and the input values.
- We call the **display_info** method on the person object using the dot notation. This will print the person's name and age to the console..

Task 3: Create a Python code that reads a file containing some initial text. Then, write to the file and append additional text to it. Observe the changes that occur in the file.

CODE :

```
# Open the file for reading
with open("My Text File.txt", "r") as file:
    content = file.read()

# Print the initial content of the file
print("\n\n\nInitial content of the file:")
print(content)

# Open the file for writing and append new text
with open("My Text File.txt", "a") as file:
    file.write("\nThis is new text appended to the file.")

# Open the file for reading again
with open("My Text File.txt", "r") as file:
    content = file.read()

# Print the final content of the file
print("\n\n\nFinal content of the file:")
print(content,"\n\n\n")
```

Output Screen Shot:

```
PS E:\Semester 4\Data Structure and Algorithm\Lab\Lab 03> python -u "e:\Semest
er 4\Data Structure and Algorithm\Lab\Lab 03\Task_03.py"



Initial content of the file:
My name is Fahim Ur Rehman Shah . This the first task
This is new text appended to the file.
This is new text appended to the file.
This is new text appended to the file.
This is new text appended to the file.
This is new text appended to the file.



Final content of the file:
My name is Fahim Ur Rehman Shah . This the first task
This is new text appended to the file.
This is new text appended to the file.
This is new text appended to the file.
This is new text appended to the file.
This is new text appended to the file.
This is new text appended to the file.



PS E:\Semester 4\Data Structure and Algorithm\Lab\Lab 03>
```

Explanation :

- We open the file named **"My Text File.txt"** in read mode using the open function and a with block. This automatically closes the file after reading is complete. We then **read** the entire contents of the file into a variable named content using the **read** method.
- We print the initial content of the file using the print function and the content variable.
- We open the file again, this time in append mode using the "a" argument to the open function. We then write new text to the file using the write method.
- We open the file for reading again using the same method as before, and read the entire contents of the file into the content variable.
- Finally, we print the final content of the file using the print function and the content variable. This will show the initial text as well as the newly appended text.

Conclusion:

In this lab session, we covered several Python programming tasks :

- Firstly, we created a Student class that takes a name and marks argument, initializes corresponding instance variables, and calculates the average of the marks list.
- Secondly, we created a Person class that takes a name and age argument, initializes corresponding instance variables, and displays the person's name and age using a method.
- Finally, we read a file containing some initial text, wrote to the file, and appended additional text to it. We observed the changes that occurred in the file by reading the file again and printing its contents.

➢ Overall, these tasks helped us to practice working with classes, methods, file input/output operations in Python, and understand how they can be used to solve common programming problems.