



Artificial Intelligence Lab

AL-2002

Lab 01

Instructor: Hurmat Hidayat
Semester: Spring 2024

Artificial Intelligence Lab 01

Objectives

The objective of this lab is to reinforce fundamental Python programming concepts and introduce basic data structures. The lab aims to provide hands-on experience in Python revision, covering essential syntax, control structures, and data structures such as lists, tuples, sets, dictionaries, and classes.

Learning Outcomes

1. Demonstrate proficiency in using basic Python syntax.
2. Implement conditional statements, loops, and user input handling in Python.
3. Design and define a class (**Student**) with attributes and methods.

Table of Contents

Objectives	1
Learning Outcomes	1
Lab Tasks	3

Lab Tasks

Task 1. Student Database Management System

You are required to implement a basic Student Database Management System using Python. The system should be capable of storing and managing student information.

Requirements:

1. Student Class:

- Create a class named **Student** with the following attributes:
 - **roll_number** (an integer)
 - **name** (a string)
 - **marks** (a dictionary where the keys are subject names and the values are corresponding marks)

2. Student Database:

- Create an empty list named **student_database** to store instances of the **Student** class.

3. Functions to Implement:

- Implement the following functions:
 - **add_student(roll_number, name, marks)**: Adds a new student to the database with the given roll number, name, and marks.
 - **display_student_details(roll_number)**: Displays the details of a student based on the roll number.
 - **display_all_students()**: Displays details of all students in the database.
 - **calculate_average_marks(roll_number)**: Calculates and displays the average marks of a student.
 - **find_topper()**: Finds and displays the details of the student with the highest average marks.

4. Menu-Driven Program:

- Create a menu-driven program that allows the user to perform the following actions:
 - Add a new student
 - Display details of a specific student
 - Display details of all students
 - Calculate and display the average marks of a student

- Find and display details of the student with the highest average marks
- Exit the program

Guidelines:

- Use loops to handle menu-driven interactions.
- Utilize appropriate data structures (lists, dictionaries) for efficient storage and retrieval of student information.