**Student Name: Student Id:**

**Lab Instructions:**Please show your work to the instructor present in the lab before submitting.

**Submission Due**: End of laboratory class, submit the file on Moodle at least 10 minutes before the end of laboratory class.

**Total Marks** = 10 marks

Marks will be given only to students who attend and participate during 2-hour laboratory class. Submission on Moodle is mandatory as an evidence of participation.

**Learning Outcomes:**

|  |  |
| --- | --- |
| L01 | Identify the basic AI concepts and its different applications. |
| L02 | Apply various AI search algorithms, and design simple models. |

**Marking Criteria:**

|  |  |  |
| --- | --- | --- |
| **Task Details** | **Submission Requirements** | **Marks** |
| **Task 1-5:** Python basics,Agents & Environment | Same **Word file** containing the source code and output. | 10 |

**Lab 1 – Basics of Python Programming**

**Introduction:**

In this lab session we will study and implement  **python exercise to** understanding basic coding techniques and its usage for development of Agents & Environments.

Programming Review

#### **Python Data Types and Structures**

* **Numbers** (Integers, Floats, Complex Numbers)
* **Strings** (String Manipulation, String Methods)
* **Lists** (List Operations, Indexing, Slicing, List Comprehension)
* **Tuples** (Immutable Sequences, Tuple Packing and Unpacking)
* **Dictionaries** (Key-Value Pairs, Dictionary Methods, Nested Dictionaries)
* **Sets** (Set Operations, Removing Duplicates)

#### **Control Flow**

* **Conditional Statements** (if, elif, else)
* **Loops** (for, while, Nested Loops)
* **Loop Control Statements** (break, continue, pass)

For an **Introduction to Artificial Intelligence (AI)** course that includes **basic Python topics**, the focus should be on **essential Python programming concepts** along with **AI-related fundamental principles**. Below is a structured breakdown of key topics that should be covered:

### **1. Python Basics for AI**

These topics provide the necessary programming foundation required for AI applications.

#### **Data Types and Structures**

* **Numbers** (Integers, Floats, Complex Numbers)
* **Strings** (String Manipulation, String Methods)
* **Lists** (List Operations, Indexing, Slicing, List Comprehension)
* **Tuples** (Immutable Sequences, Tuple Packing and Unpacking)
* **Dictionaries** (Key-Value Pairs, Dictionary Methods, Nested Dictionaries)
* **Sets** (Set Operations, Removing Duplicates)

#### **Control Flow**

* **Conditional Statements** (if, elif, else)
* **Loops** (for, while, Nested Loops)
* **Loop Control Statements** (break, continue, pass)

#### **Functions**

* **Defining Functions** (def keyword)
* **Function Arguments** (Positional, Keyword, Default Arguments, \*args, \*\*kwargs)
* **Lambda Functions** (Anonymous Functions lamda key word)

#### **Object-Oriented Programming (OOP)**

* **Classes and Objects** (Creating Objects, self keyword)
* **Methods and Attributes** (Instance and Class Attributes)
* **Encapsulation, Inheritance, Polymorphism**
* **Magic Methods (\_\_init\_\_, \_\_str\_\_, etc.)**

#### **File Handling**

* **Reading and Writing Files (.txt, .csv, .json)**
* **Handling Exceptions (try, except, finally)**

**Practice Task 1:**

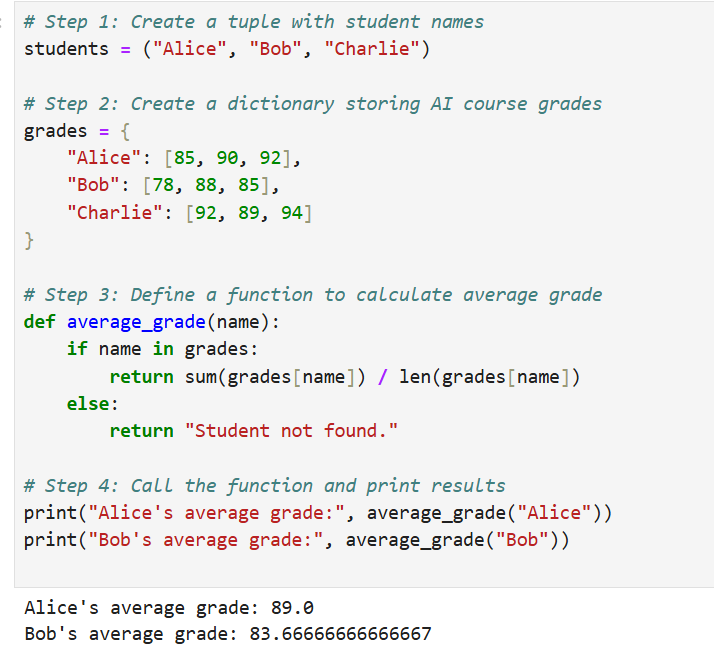
A university wants to store AI course grades for students using **tuples and dictionaries**.

* Create a **tuple** named students containing three names: "Alice", "Bob", and "Charlie".
* Create a **dictionary** named grades where:
  + Keys are student names.
  + Values are **lists** of three AI course grades.

Write a function average\_grade(name) that **returns the average grade** for a student. Call the function with "Alice" and print the result.

**Task :** The sample code snippet given below , write your on code and replace with your name and your friend name in the code and print your average mark

Sample code –

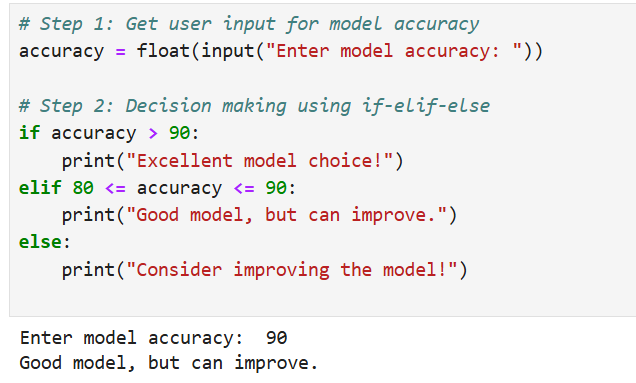


**Student Code and output**

**Practice Task 2:**

A researcher is selecting an AI model based on **accuracy**. Write a Python program that:

* Asks the user to **input** a model accuracy percentage (float).
* **Prints a recommendation** based on accuracy:
  + **Above 90%** → "Excellent model choice!"
  + **Between 80-90%** → "Good model, but can improve."
  + **Below 80%** → "Consider improving the model!"
* Use if-elif-else statements.



**Task :** The sample code snippet given above , write your own code and add one more input as model name and print the model name in the output

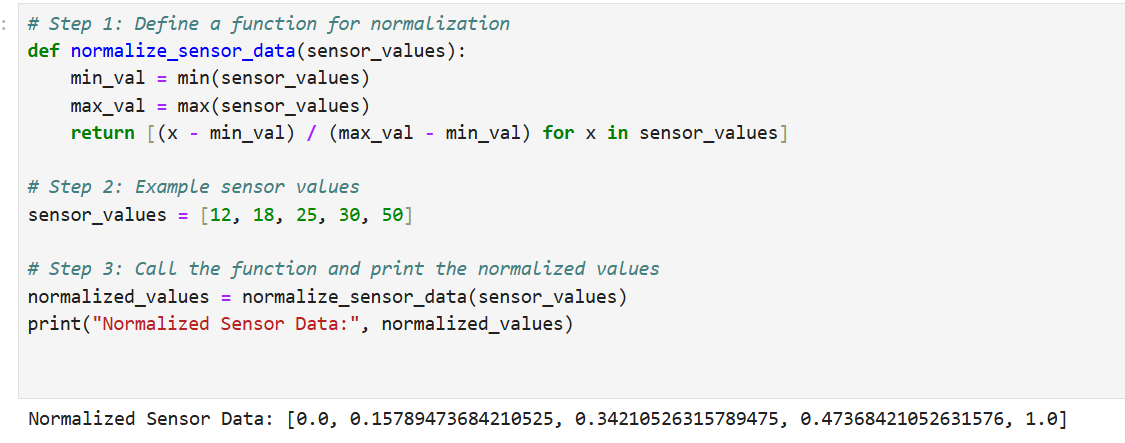
**Practice Task 3:**

Write a function normalize\_sensor\_data(sensor\_values) that:

* Takes a **list** of sensor readings (floats).
* Normalizes them between **0 and 1** using the formula: Xnormalized=(X−Xmin)/(Xmax−Xmin)
* Returns the **normalized list**.

**Task :** The sample code snippet given below, write your own code and normalise the below data and get your normalisation result

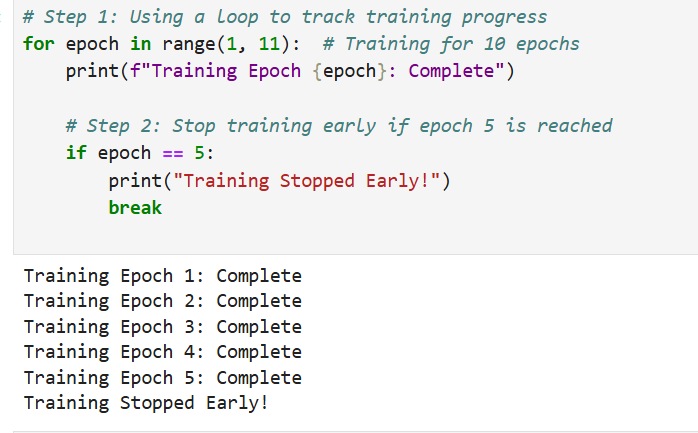
Test data : 1,5,35, 110, 2000, 8500, 70



**Practice Task 4:**

You are training a machine learning model for **10 epochs**.Write a Python program that:

* Uses a **for loop** to simulate **10 training epochs**.
* Prints "Training Epoch X: Complete" where X is the epoch number.
* Stops training if epoch **5** is reached using break.

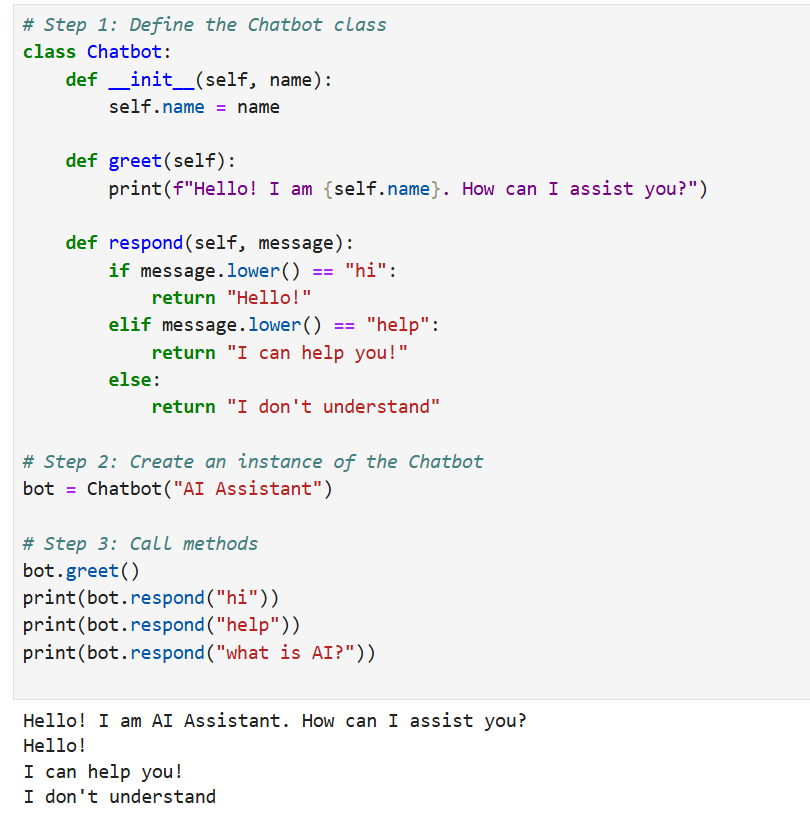


**Task :** The sample code snippet given above, write your own code and implement same with while loop

**Task 5:**

Create a **class Chatbot** that:

* Has an \_\_init\_\_ method to store a **bot name**.
* Has a method greet() that **greets the user**.
* Has a method respond(message) that:
  + Returns "Hello!" if the user types "hi".
  + Returns "I can help you!" if the user types "help".
  + Returns "I don't understand" for other messages.



**Task :** The sample code snippet given above, write your own code and add one more attribute year of birth and one more method age for calculating current age.

Example : YOB=2015, Age=2025-YOB

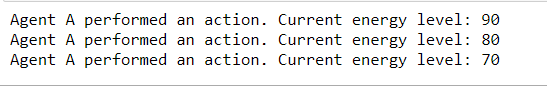
**Basic Agent Class Creation**

**Objective:** Create a simple agent class that can perform a basic action.

**Instructions:**

1. Create a class named SimpleAgent.
2. This class should have an \_\_init\_\_ method that initializes the agent's name and energy level.
3. Add a method perform\_action that decreases the agent's energy by 10 each time it's called and prints a message with the agent's name and current energy level.
4. Instantiate the SimpleAgent class with the name "Agent A" and an energy level of 100.
5. Call the perform\_action method three times and observe the output.

**Sample Output**



**Student Code and output**

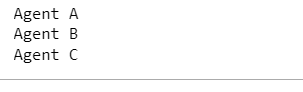
**Lab Task 2: Creating an Environment Class**

**Objective:** Create an environment class where agents can interact.

**Instructions:**

1. Create a class named Environment.
2. The class should have an \_\_init\_\_ method that initializes an empty list to keep track of agents.
3. Add a method add\_agent that adds an agent to the environment.
4. Add another method get\_agents that returns the list of all agents currently in the environment.
5. Instantiate the Environment class, add three different agents to it, and print out the list of agents.

**Sample Output:**



**Student Code and Output:**

**Submission Instructions:**

1. Submit your answers in this word file by renaming it in the format **“INTROAI\_** **AI2010\_Lab\_1\_03\_10\_2024\_name”** and uploading on **Moodle** in the appropriate submission link.
2. Also submit source code (.ipynb) files for Lab Tasks along with this word file.

**Please conform to the naming convention of the file.**

**END OF LABORATORY**