National University of computer and emerging sciences

Artificial Intelligence Lab (AI-2002)

Semester: Spring 2026
Section: BCS-6A
Course Instructor: Mr. Abdullah Shaikh

# LAB # 02 – IN-LAB TASKS

## Getting Started with Object-Oriented Programming (OOP) with Python

Submission Instructions:

- Perform all tasks in class only. Late submissions will not be considered.
- Submit a .ipynb file with the filename formatted as:
  Lab 02 - Your Name - Roll Number
- Upload the file on GCR.

### Task # 01

Write a program in the Python programming language to create a class named Student.
The class should have two data members: name and roll_number.
Define a member function display() that prints the values of these data members.

Create an object of the Student class, assign values to the data members using the object, and display the output.

**Sample Input**

Name: Ali
Roll Number: 101

**Sample Output**

Name: Ali
Roll Number: 101

## Task # 02

Write a Python program to manage students and their marks using classes.

1. Create a base class Student with the following:
   - Data members: name, roll_number
   - Constructor to initialize these data members
   - Method display() to print student details
2. Create a derived class Exam that inherits from Student:
   - Additional data member: marks (a list of integers)
   - Constructor should call the parent constructor using super()
   - Method average() to calculate and return the average marks
   - Method display_result() to print the student details and their average marks
3. Create two objects of the Exam class with different data, and display the results.

**Sample Input**

For Object 1:

Name: Ali
Roll Number: 101
Marks: 80 90 85

For Object 2:

Name: Sara
Roll Number: 102
Marks: 75 85 95

**Sample Output**

Student Details:
Name: Ali
Roll Number: 101
Average Marks: 85.0

Student Details:
Name: Sara
Roll Number: 102
Average Marks: 85.0

Task # 03

Write a Python program to create a class named BankAccount that demonstrates private members.

1. The class should have the following private data members:
   - __account_number
   - __balance
2. Create a constructor to initialize the account number and balance.
3. Create the following public methods:
   - deposit(amount) → adds amount to balance
   - withdraw(amount) → subtracts amount from balance if sufficient funds are available, otherwise prints "Insufficient balance"
   - display() → prints the account number and balance
4. Create an object of the class, perform some deposits and withdrawals, and display the final account details.

**Sample Input**

Account Number: 12345
Initial Balance: 5000
Deposit: 2000
Withdraw: 3000

**Sample Output**

Account Number: 12345
Balance: 4000

Task # 04

Write a Python program to manage different types of employees in a company using method overriding.

1. Create a base class Employee with:
   - Data members: name, salary
   - Constructor to initialize name and salary
   - Method get_details() → prints the employee's name and salary
2. Create a derived class Manager that inherits from Employee:
   - Additional data member: bonus
   - Constructor should call the parent constructor using super()
   - Override the get_details() method to also include the bonus and total compensation (salary + bonus)
3. Create a derived class Developer that inherits from Employee:
   - Additional data member: project
   - Constructor should call the parent constructor using super()
   - Override the get_details() method to also include the project name
4. Create one object of Manager and one object of Developer, and call get_details() for each.

**Sample Input**

For Manager:
Name: Ahmed
Salary: 80000
Bonus: 15000

For Developer:
Name: Sara
Salary: 60000
Project: AI Chatbot

**Sample Output**
Manager Details:
Name: Ahmed
Salary: 80000
Bonus: 15000
Total Compensation: 95000

Developer Details:
Name: Sara
Salary: 60000
Project: AI Chatbot

THE END!