



Artificial Intelligence

Lab 05 Tasks

Name: Faareha Raza

Sap ID: 47431

Batch: BSCS-6th semester

Lab Instructor:

Ayesha Akram

Class Tasks:

1. Define a function that accepts roll number and returns whether the student is present or absent.

```
Lab Task5.py  Lab Task1.py X
AI LAB5 > Lab Task1.py > ...
1  attendance = {}
2  for i in range(3):
3      roll_number = int(input("Enter roll number: "))
4      status = input("Enter status (Present/Absent): ")
5      attendance[roll_number] = status
6  def check_attendance(roll_number):
7      if roll_number in attendance:
8          return attendance[roll_number]
9      else:
10         return 'Roll number not found'
11 roll_number = int(input("Enter roll number to check: "))
12 status = check_attendance(roll_number)
13 print(f"Roll number {roll_number} is {status}.")
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  COMMENTS
● PS C:\AI Lab> python -u "c:\AI Lab\AI LAB5\Lab Task1.py"
Enter roll number: 47431
Enter status (Present/Absent): Present
Enter roll number: 57643
Enter status (Present/Absent): Absent
Enter roll number: 78654
Enter status (Present/Absent): Present
Enter roll number to check: 47431
Roll number 47431 is Present.
❖ PS C:\AI Lab> 
```

2. Define a class and create multiple object of class, access attributes and assign new values.

```
AI LAB5 > Lab Task2.py > ...
1 class Car:
2     def __init__(self, make, model, year):
3         self.make = make
4         self.model = model
5         self.year = year
6
7     def display_info(self):
8         print(f"Car: {self.year} {self.make} {self.model}")
9 # Create multiple objects of the Car class
10 car1 = Car("Toyota", "Corolla", 2020)
11 car2 = Car("Honda", "Civic", 2019)
12 car3 = Car("Ford", "Mustang", 2021)
13 # Access attributes
14 print(car1.make)
15 print(car2.model)
16 print(car3.year)
17 car1.year = 2022
18 car2.make = "Hyundai"
19 car3.model = "Explorer"
20 car1.display_info()
21 car2.display_info()
22 car3.display_info()
```

```
PS C:\AI Lab> python -u "c:\AI Lab\AI LAB5\Lab Task2.py"
Toyota
Civic
2021
Car: 2022 Toyota Corolla
Car: 2019 Hyundai Civic
Car: 2021 Ford Explorer
PS C:\AI Lab>
```

3. Create a student class with attributes name, age, and grades (list). Add a method average grade that calculates and returns the average of the grades.

```
AI LAB5 > Lab Task3.py > ...
1 class Student:
2     def __init__(self, name, age, grades):
3         self.name = name
4         self.age = age
5         self.grades = grades
6
7     def average_grade(self):
8         return sum(self.grades) / len(self.grades) if self.grades else 0
9 student = Student("Fareha Raza", 20, [85, 90, 78, 92])
10 print(f"Average grade: {student.average_grade()}")
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENT
● PS C:\AI Lab> python -u "c:\AI Lab\AI LAB5\Lab Task3.py"
Average grade: 86.25
● PS C:\AI Lab> python -u "c:\AI Lab\AI LAB5\Lab Task3.py"
```

4. Create a base class **Employee** with:

- name
- salary
- Method **display_details()** to show employee info.

Create two subclasses:

1. **Manager** (inherits Employee) and has an additional attribute **department**
2. **Developer** (inherits Employee) and has an additional attribute **programming_language**

Override the **display_details()** method in both subclasses to include their specific attributes.

```
Lab Task3.py Lab Task1.py Lab Task2.py Lab Task3.py Lab Task4.py
LAB5 > Lab Task4.py > Developer > display_details
1 # Base class Employee
2 class Employee:
3     def __init__(self, name, salary):
4         self.name = name
5         self.salary = salary
6
7     def display_details(self):
8         print(f"Name: {self.name}")
9         print(f"Salary: {self.salary}")
10
11 # Subclass Manager
12 class Manager(Employee):
13     def __init__(self, name, salary, department):
14         super().__init__(name, salary)
15         self.department = department
16
17     def display_details(self):
18         super().display_details()
19         print(f"Department: {self.department}")
20
21 # Subclass Developer
22 class Developer(Employee):
23     def __init__(self, name, salary, programming_language):
24         super().__init__(name, salary)
25         self.programming_language = programming_language
26
27     def display_details(self):
28         super().display_details()
29         print(f"Programming Language: {self.programming_language}")
30
31 manager = Manager("Faareha", 75000, "IT")
32 developer = Developer("Raza", 65000, "Python")
33
34 print("Manager Details:")
35 manager.display_details()
36
37 print("\nDeveloper Details:")
38 developer.display_details()
```

```
PS C:\AI Lab> python -u "c:\AI Lab\AI LAB5\Lab Task4.py"
Manager Details:
Name: Faareha
Salary: 75000
Department: IT

Developer Details:
Name: Raza
Salary: 65000
Programming Language: Python
PS C:\AI Lab>
```

5. Create a base class **Shape** with a method **area()**.

Create the following subclasses:

- **Circle** (takes radius and implements area() as $\pi * r^2$)
- **Rectangle** (takes length and width and implements area() as $\text{length} \times \text{width}$)
- **Triangle** (takes base and height and implements area() as $0.5 \times \text{base} \times \text{height}$)

Use **polymorphism** to calculate the area of different shapes.

```
AI LAB5 > Lab Task5.py > Triangle
1  import math
2
3  class Shape:
4      def area(self):
5          pass
6
7  class Circle(Shape):
8      def __init__(self, radius):
9          self.radius = radius
10
11      def area(self):
12          return math.pi * self.radius ** 2
13
14  class Rectangle(Shape):
15      def __init__(self, length, width):
16          self.length = length
17          self.width = width
18
19      def area(self):
20          return self.length * self.width
21
22  class Triangle(Shape):
23      def __init__(self, base, height):
24          self.base = base
25          self.height = height
26
27      def area(self):
28          return 0.5 * self.base * self.height
29
30  shapes = [
31      Circle(5),
32      Rectangle(4, 6),
33      Triangle(3, 7)
34  ]
35
36  for shape in shapes:
37      print(f'The area of the shape is: {shape.area()}')
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  COMMENTS

• PS C:\VAI Lab> python -u "c:\VAI Lab\VAI LAB5\Lab Task5.py"
• PS C:\VAI Lab> python -u "c:\VAI Lab\VAI LAB5\Lab Task5.py"
  The area of the shape is: 78.53981633974483
  The area of the shape is: 24
  The area of the shape is: 10.5
❖ PS C:\VAI Lab>
```