Artificial Intelligence

Lab 05 Tasks

Name: Dua Amir

Sap ID: 47849

Batch: BSCS-6th semester

Lab Instructor:

Ayesha Akram

Task1.

Solution:

```python
def check_attendance(roll_number, present_students):  1 usage
    if roll_number in present_students:
        return "Present"
    else:
        return "Absent"


present_students_list = [47849, 46462, 46484, 47789, 43507]
roll_no = int(input("Enter roll number: "))
attendence = check_attendance(roll_no, present_students_list)
print(f"Student with roll number {roll_no} is {attendence}.")
```

```
Enter roll number: 47849
Student with roll number 47849 is Present.
```

Task2.
Solution:

```python
class Student:  2 usages
    def __init__(self, roll_number, name):
        self.roll_number = roll_number
        self.name = name
    def details(self):  4 usages
        print(f"Roll Number: {self.roll_number}, Name: {self.name}")
student1 = Student( roll_number: 46462,  name: "Zainab")
student2 = Student( roll_number: 46484,  name: "Samreen")
print("Before updating:")
student1.details()
student2.details()
student1.name = "Dua"
student2.roll_number = 47849
print("After updating:")
student1.details()
student2.details()
```

```
Before updating:
Roll Number: 46462, Name: Zainab
Roll Number: 46484, Name: Samreen
After updating:
Roll Number: 46462, Name: Dua
Roll Number: 47849, Name: Samreen
```

Task3.
Solution:

```python
class Student:  1 usage
    def __init__(self, name, age, grades):
        self.name = name
        self.age = age
        self.grades = grades

    def average_grade(self):  1 usage
        return sum(self.grades) / len(self.grades)

student1 = Student( name: "Dua", age: 20, grades: [85, 90, 87, 92, 88])

print("Name:", student1.name)
print("Age:", student1.age)
print("Average Grade:", student1.average_grade())
```

```
Name: Dua
Age: 20
Average Grade: 88.4
```

Task4.
Solution:

```python
class Employee:  2 usages
    def __init__(self, name, salary):
        self.name = name
        self.salary = salary
    def display_details(self):
        print(f"Name: {self.name}, Salary: {self.salary}")
class Manager(Employee):  1 usage
    def __init__(self, name, salary, department):
        super().__init__(name, salary)
        self.department = department
    def display_details(self):  1 usage
        print(f"Name: {self.name}, Salary: {self.salary}, Department: {self.department}")
class Developer(Employee):  1 usage
    def __init__(self, name, salary, programming_language):
        super().__init__(name, salary)
        self.programming_language = programming_language
    def display_details(self):  1 usage
        print(f"Name: {self.name}, Salary: {self.salary}, Programming Language: {self.programming_language}")
manager1 = Manager( name: "Dua", salary: 150000, department: "HR")
developer1 = Developer( name: "Ayesha", salary: 80000, programming_language: "Python")
manager1.display_details()
developer1.display_details()
```

```
Name: Dua, Salary: 150000, Department: HR
Name: Ayesha, Salary: 80000, Programming Language: Python
```

Task5.
Solution:

```python
import math
class Circle:  1 usage
    def __init__(self, radius):
        self.radius = radius
    def area(self):  1 usage
        return math.pi * self.radius ** 2
class Rectangle:  1 usage
    def __init__(self, length, width):
        self.length = length
        self.width = width
    def area(self):  1 usage
        return self.length * self.width
class Triangle:  1 usage
    def __init__(self, base, height):
        self.base = base
        self.height = height
    def area(self):  1 usage
        return 0.5 * self.base * self.height
print("Circle Area:", Circle(5).area())
print("Rectangle Area:", Rectangle( length: 4,  width: 6).area())
print("Triangle Area:", Triangle( base: 3,  height: 8).area())
```

```
Circle Area: 78.53981633974483
Rectangle Area: 24
Triangle Area: 12.0
```

_____