Artificial Intelligence

Lab 05

Name: Hajira Imran

Sap ID: 44594

Submitted to: Ma'am Ayesha Akram

Batch: BSCS-6th semester

Task 01

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

task 01.py

```python
def check_attendance(roll_number):  1 usage
    present_students = {101, 102, 103, 105}

    return "Present" if roll_number in present_students else "Absent"

roll_number = int(input("Enter roll number: "))
print(check_attendance(roll_number))
```

Output

Run task 01

```
C:\Users\lenovo\PycharmProjects\PythonProject3\.venv\Scripts\python.exe "C:\Users\lenovo\Py
Enter roll number: 103
Present

Process finished with exit code 0
```

Task 02

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

task 01.py | task 02.py | task 03.py | task 04.py | task 05.py

```
class Car:  3 usages
    def __init__(self, brand, model, year, price):
        self.model = model
        self.year = year
        self.price = price

    def display_info(self):  3 usages
        print(f"Car: {self.brand} {self.model}, Year: {self.year}, Price: {self.price}")

# Objects
car1 = Car( brand: "Toyota",  model: "Corolla",  year: 2022,  price: 3000000)
car2 = Car( brand: "Honda",  model: "Civic",  year: 2023,  price: 3500000)
car3 = Car( brand: "Ford",  model: "Mustang GT (V8)",  year: 2024,  price: 10000000)

# Access attributes
print(car1.brand)
print(car2.year)
print(car3.price)

# Modify attributes
car1.year = 2025
car2.model = "Accord"
car3.price = 12000000

car1.display_info()
car2.display_info()
car3.display_info()
```

Output

Task 03

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

task 01.py | task 02.py | task 03.py | task 04.py | task 05.py

```python
class Student:  3 usages
    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) if self.grades else 0

    def display_info(self):  3 usages
        print(f"Student: {self.name}, Age: {self.age}, Average Grade: {self.average_grade():.2f}")

# Create student objects
student1 = Student( name: "Ali",  age: 20,  grades: [85, 90, 78, 92])
student2 = Student( name: "Ayesha",  age: 22,  grades: [88, 76, 95, 89])
student3 = Student( name: "Hassan",  age: 19,  grades: [90, 85, 87, 80])

# Access attributes
print(student1.name)
print(student2.age)
print(student3.grades)

# Display student info
student1.display_info()
student2.display_info()
student3.display_info()
```

Output

```
Ali
22
[90, 85, 87, 80]
Student: Ali, Age: 20, Average Grade: 86.25
Student: Ayesha, Age: 22, Average Grade: 87.00
Student: Hassan, Age: 19, Average Grade: 85.50

Process finished with exit code 0
```

Task 04

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.

task 01.py | task 02.py | task 03.py | task 04.py | task 05.py

```python
# Parent class
class Employee:  3 usages
    def __init__(self, name, salary):
        self.name = name
        self.salary = salary

    def display_details(self):  1 usage
        print(f"Employee Name: {self.name}, Salary: {self.salary}")

# Subclass 1: Manager
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"Manager Name: {self.name}, Salary: {self.salary}, Department: {self.department}")

# Subclass 2: Developer
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"Developer Name: {self.name}, Salary: {self.salary}, Programming Language: {self.programming_language}")

#objects for each class
emp = Employee( name: "John Doe",  salary: 50000)
mgr = Manager( name: "Alice Smith",  salary: 80000,  department: "Sales")
dev = Developer( name: "Bob Johnson",  salary: 70000,  programming_language: "Python")
emp.display_details()
mgr.display_details()
dev.display_details()
```

Output

Run | task 04

```
C:\Users\lenovo\PycharmProjects\PythonProject3\.venv\Scripts\python.exe "C:\Users\lenovo\PycharmProjects\PythonProject3\task 04.py"
Employee Name: John Doe, Salary: 50000
Manager Name: Alice Smith, Salary: 80000, Department: Sales
Developer Name: Bob Johnson, Salary: 70000, Programming Language: Python

Process finished with exit code 0
```

Task 05

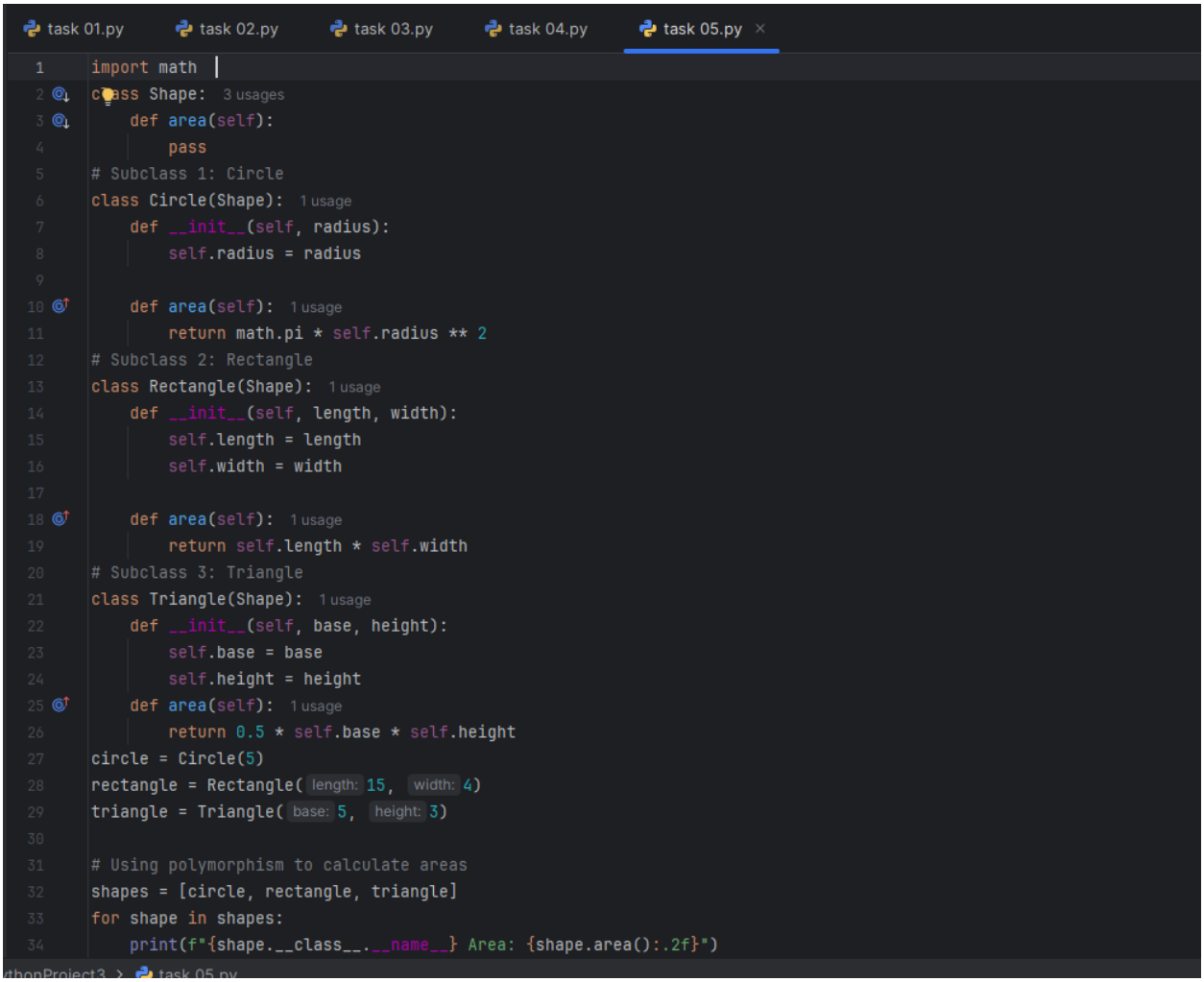
Create a base class Shape with a method area().

Create the following subclasses:

- Circle (takes radius and implements area() as $\pi * r^2$)
- Rectangle (takes length and width and implements area() as length × width)

- Triangle (takes base and height and implements area() as 0.5 × base × height)

Use polymorphism to calculate the area of different shapes.



Output


Run
task 05
C:\Users\lenovo\PycharmProjects\PythonProject3\.venv\Scripts\python.exe "C:\Users\lenovo\Pyc
Circle Area: 78.54
Rectangle Area: 60.00
Triangle Area: 7.50
Process finished with exit code 0