OBJECT ORIENTED PROGRAMMING 2 LABORATORY
Experiment # 7:
Inheritance and Polymorphism

OBJECTIVES

The main purpose of this experiment is to introduce you to inheritance and polymorphism concepts. In this experiment, firstly, inheritance and polymorphism are examined. Then, some examples are studied.

QUESTIONS

1) We have examined inheritance concept in the course. Now, we will implement the following UML Class Diagram which is given in Figure 1. In the diagram, the Number Class is superclass and the Square and the Cube classes are derived from the Number superclass.

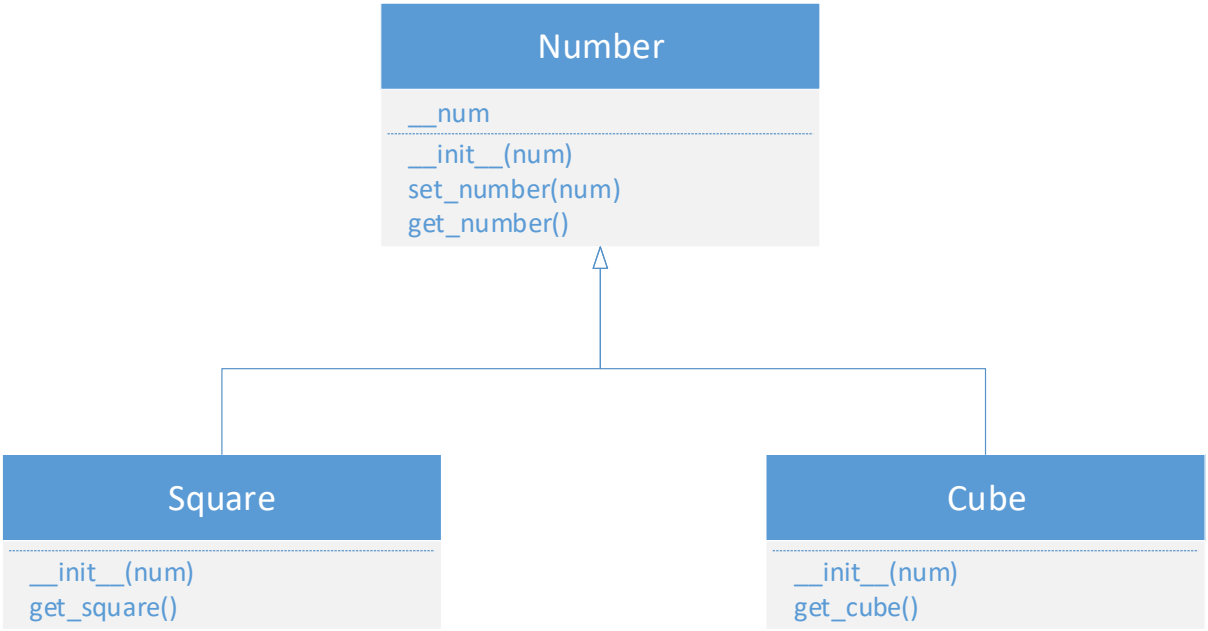

Fig 1. UML Class Diagram for Number, Square and Cube Classes

Test your program with following driver program.

```
import num

def main():

    my_square = num.Square(5)
    my_cube = num.Cube(5)

    print("Square is ", my_square.get_square())
    print("Cube is ", my_cube.get_cube())

main()
```

2) Repeat Question 1, for the following UML Class Diagram.

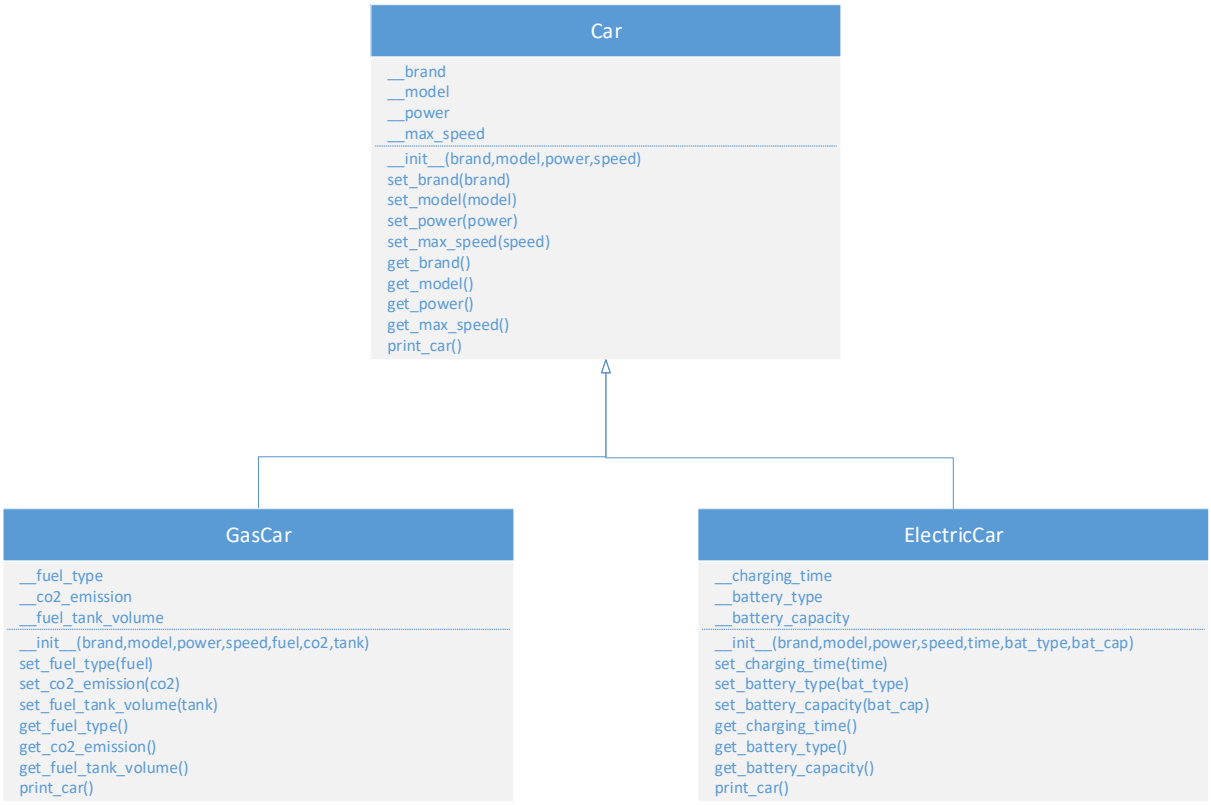

Fig 2. UML Class Diagram for Car, GasCar and ElectricCar Classes

Car Class

- In __init__ method, initialize data attributes and print the brand and model of the object.
- In print_car method, print the data attributes according to the output which is given below. Use accessor methods in print_car method.

GasCar Class

- In __init__ method, call car class' __init__ method to initialize data attributes of the superclass (brand, model, power and max_speed).
- In print_car method, call base class's print_car method. Then, print data attributes of the GasCar Class. Use accessor methods in print_car method.

ElectricCar Class

- In __init__ method, call car class' __init__ method to initialize data attributes of the superclass (brand, model, power and max_speed).
- In print_car method, call base class's print_car method. Then, print data attributes of the ElectricalCar Class. Use accessor methods in print_car method.

Test your program with following driver program.

```
import cars

def main():

    my_gas_car = cars.GasCar("Volkswagen","Golf", 115,198,"Diesel",109,50)
    my_gas_car.print_car()

    my_electrical_car=cars.ElectricCar("Tesla","Model S", 301,193,20,"Li-ion",60)
    my_electrical_car.print_car()

main()
```

The output of the program must be as follows:

```
Initializer for Volkswagen Golf
Brand: Volkswagen
Model: Golf
Power: 115 HP
Max Speed: 198 km/h
Fuel Type: Diesel
CO2 Emission: 109 g/km
Fuel Tank Volume: 50 L

Initializer for Tesla Model S
Brand: Tesla
Model: Model S
Power: 301 HP
Max Speed: 193 km/h
Charging Time: 20 hours
Battery Type:  Li-ion
Battery Capacity:  60 kW/h
```

3) Develop a polymorphic calculator program. Design a class for real numbers. This class will be superclass. The real class will have two private attributes. The class must have appropriate __init__, accessor and mutator methods. Then, the class must have add and subtract methods. Then, design a class for complex numbers. This class will be subclass of real superclass. The complex class will have four private attributes. The class must have appropriate __init__, accessor and mutator methods. Then, the class must have add and subtract methods.

Test your program with following driver program.

```
import calculator

def main():

    my_real=calculator.Real(3,4)
    print(my_real.add())
    print(my_real.substract())

    my_complex=calculator.Complex(0,8,0,2)
    my_complex.set_num1(7)
    my_complex.set_num2(1)
    print(my_complex.add())
    print(my_complex.substract())

main()
```