

OBJECT ORIENTED PROGRAMMING I LABORATORY**Experiment # 11:****Inheritance****QUESTIONS**

1) We have examined the 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 base classes and the Square and the Cube classes are derived from the Number base Class.

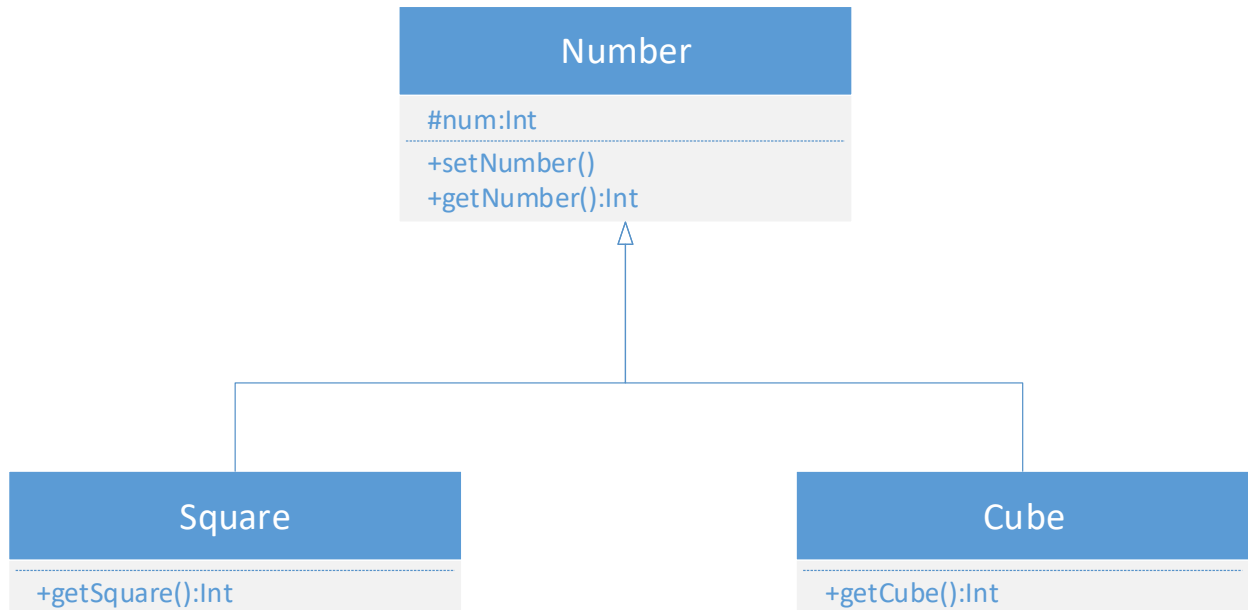


Fig 1. UML Class Diagram for Number, Square and Cube Classes

Number Class

- Define a **protected int** data member (i.e., num)
 - Include setNumber function, in the function; **prompt the number from the user.**
- i) Write a C++ code to implement the UML class diagram. You can write class definition and main function in the same file (in a .cpp file). Also, you can take advantage of the protected access specifier when you use data members of the base class within derived classes.
- ii) Repeat Question i), **by changing protected** access specifier with **private** access specifier for base class' data member. Make required modifications to handle that change.
- iii) Repeat Question ii) while separating base and derived classes into header and implementation files.

Test your program with the following driver program.

```

8  #include <iostream>
9  #include "Square.h"
10 #include "Cube.h"
11 using namespace std;
12
13 main()
14 {
15     Square objS;
16     Cube objC;
17     int sqr,cube;
18
19     objS.setNumber();
20     sqr=objS.getSquare();
21     cout << "Square of "<< objS.getNumber() << " is: " << sqr << endl;
22
23     objC.setNumber();
24     cube=objC.getCube();
25     cout << "Cube of "<< objS.getNumber() << " is: " << cube << endl;
26
27 }
```

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

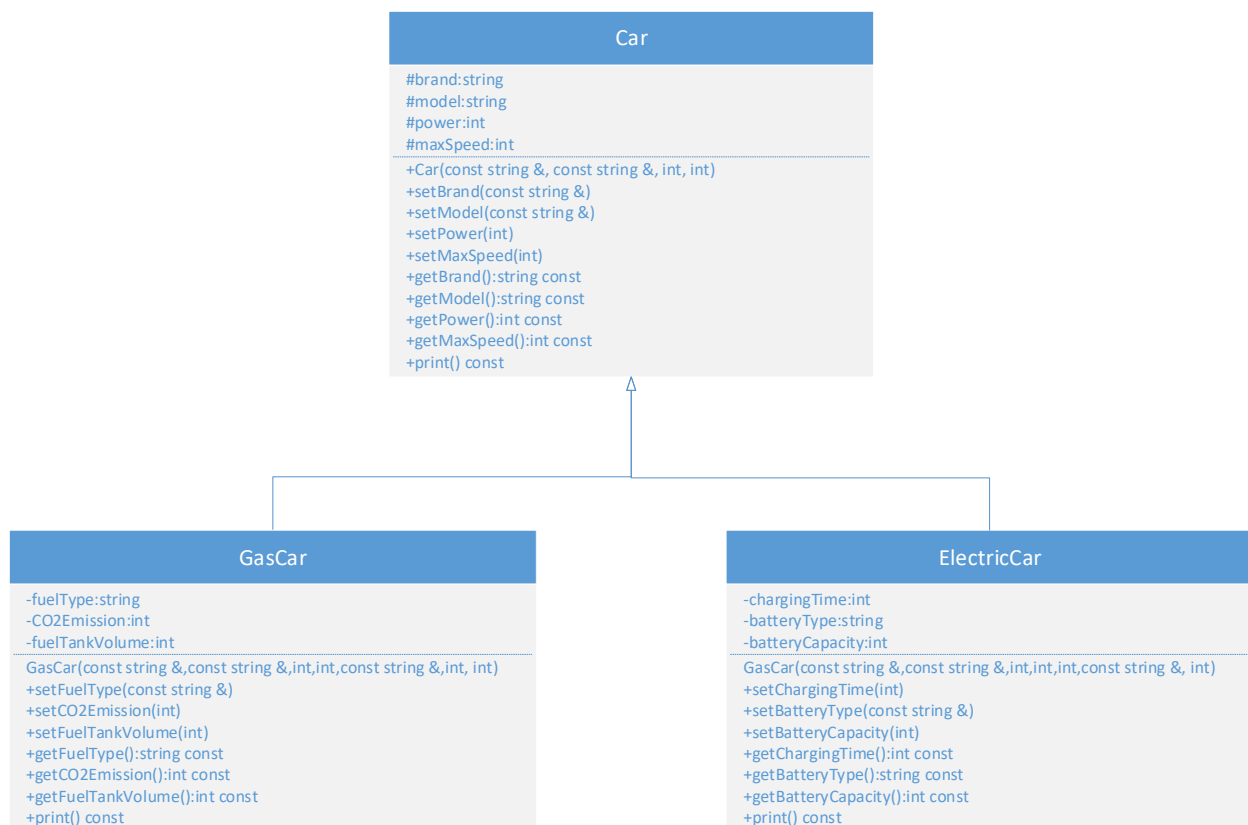


Fig 2. UML Class Diagram for Car, GasCar, and ElectricCar Classes

Car Class

- In constructor, use member initializer (:) to initialize data members and print the brand and model of the object.
- In print function, print the data members according to the output, which is given below.

GasCar Class

- In constructor, use **base-class initializer syntax** to initialize data members of the base class (brand, model, power, and maxSpeed). Then, use set functions to set initial values of the GasCar class data members (fuelType, CO2Emission, fuelTankVolume). (**Hint:** You can get help from your textbook at pages 515-517 [1]).
- In print function, call the base class's print function. Then print data members of the GasCar Class. (**Hint:** You can get help from your textbook at pages 525-526 [1]).

ElectricCar Class

- In constructor, use **base-class initializer syntax** to initialize data members of the base class (brand, model, power, and maxSpeed). Then, use set functions to set initial values of the ElectricCar class data members (chargingTime, batteryType, batteryCapacity). (**Hint:** You can get help from your textbook at pages 515-517 [1]).
- In print function, call base class's print function. Then print data members of the ElectricalCar Class. (**Hint:** You can get help from your textbook at pages 525-526 [1]).

Test your program with the following driver program.

```

8  #include <iostream>
9  #include "GasCar.h"
10 #include "ElectricCar.h"
11 using namespace std;
12
13 main()
14 {
15     // instantiate a GasCar object
16     GasCar c1("Volkswagen", "Golf", 115, 198, "Diesel", 109, 50);
17     c1.print();
18
19     // instantiate a ElectricCar object
20     ElectricCar c2("Tesla", "Model S", 301, 193, 20, "Li-ion", 60);
21     c2.print();
22 }
23

```

The output of the program must be as follows:

```

Constructor 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

Constructor for Tesla Model S
Brand: Tesla
Model: Model S
Power: 301 HP
Max Speed: 193 km/h
Charging Time: 20 hour
Battery Type: Li-ion
BatteryCapacity: 60 kW/h

```

[1] Paul Deitel and Harvey Deitel, "C++ How To Program", Eighth Edition, Prentice Hall, 2012.