

Exercise 5

1. จงติดตามการทำงานของโปรแกรมข้างล่างนี้ว่ามีที่ผิดที่ใดบ้าง จงอธิบาย

```
#include<iostream>
```

```
using namespace std;
```

```
class X {  
    int a;  
    protected:  
    int b;  
    public:  
    X() { cout<<"help me"<<endl; }  
    X(int a, int b) { a=1; b=2; }  
    void set(int t) { a=t; cout<<"ie ie ... Wrong"<<endl; }  
};
```

```
class Y: public X {  
    static int s;  
    protected:  
    int t;  
    public:  
    Y() { a=1; b=2; t=s++; }  
    void set(X x) {  
        x.b=t;  
        cout<<"Oh no..."<<s<<endl;  
    }  
};
```

```
class Z: Y {  
    int d;  
    public:  
    void h() { t=2; }  
    void g(Y y) { y.t=1; }  
    friend void f(Y y) { y.t=0; }  
};
```

```
int Y::s;
```

```

void main() {
    Y y;
    X x(1,2);
    y.set(x);
    Z z;
    z.set(x);
    z.set(5);
}

```

2. จงติดตามการทำงานของโปรแกรมข้างล่างนี้ว่าได้ผลลัพธ์อะไรบ้าง

```

#include<iostream>
using namespace std;

```

```

class A {
    int x;
public:
    A() {x=1; cout<<'A'<<endl; }
    A(int a) { x=a; }
    ~A() { cout<<x<<endl; }
    A f() {return *this;}
    void g() { cout<<"good"<<endl; }
    void show() { cout<<"who am I"<<endl; }
};

```

```

class B: public A {
    int y;
public:
    B() { y=2; cout<<'B'<<endl; }
    B(int a, int b):A(b) { y=a; cout<<a+b<<endl; }
    ~B() { cout<<y<<endl; }
    B f() {return B(2,3); }
    void g() { cout<<"great"<<endl; }
};

```

```

class C: public B {
    int z;
public:
    C() { z=1; cout<<'C'<<endl; }
    ~C() { cout<<z<<endl; }
};

```

```

class D:public B {
public:
    D() { cout<<'D'<<endl; }
    D(int k):B(2,1) { cout<<k<<endl; }
    ~D() { cout<<"I am D"<<endl; }
};

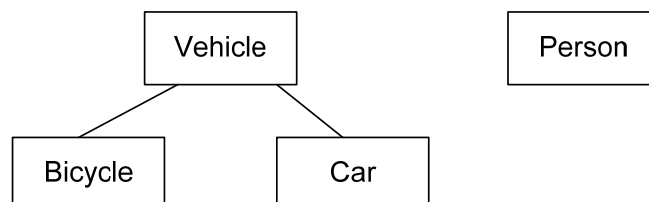
```

```

void main() {
    A a;
    B b(1,3);
    C *c;
    D d(1);
    B *e = new D;
    e->f().g();
    a.f().g();
    b=d.f();
    b.show();
}

```

5.



จงเขียนโปรแกรมภาษา C++ ในการสร้าง class Vehicle, class Bicycle, class Car และ class Person โดยที่ class ทั้งสี่มีความสัมพันธ์ดังแผนภาพที่แสดงข้างบนนี้โดยที่ Bicycle และ Car สืบทอดมรดกแบบ public มาจาก class Vehicle โดยแต่ละ class มีการเก็บข้อมูลต่างๆในส่วนของ private ดังรายละเอียดต่อไปนี้

class Person ประกอบด้วยข้อมูล	char name[20]	ชื่อ
	char surname[20]	นามสกุล
	int numCar	จำนวนรถยนต์ที่เป็นเจ้าของ
	Vehicle *myCar	รถยนต์ทั้งหมดที่เป็นเจ้าของ

class Vehicle ประกอบด้วยข้อมูล	int vID	รหัสยานพาหนะ
class Bicycle ประกอบด้วยข้อมูล	int noSeat	จำนวนที่นั่ง
class Car ประกอบด้วยข้อมูล	int type	รหัสชนิดของรถยนต์ เช่น รหัส 1 คือ Economy 2 คือ Compact 3 คือ Intermediate 4 คือ Full size 5 คือ Minivan เป็นต้น

จงสร้างฟังก์ชันในส่วนของ public ของแต่ละคลาสตามเงื่อนไขต่างๆ ดังนี้

1. จงสร้างฟังก์ชันในการนับจำนวนอีอบเจกต์ของ Car ทั้งหมดที่ถูกสร้างขึ้นและคงอยู่ในโปรแกรม ในกรณีที่ไม่มี Car ถูกสร้างขึ้นเลยก็ต้องสามารถแสดงได้ว่าไม่มีอีอบเจกต์ของ Car ใดๆถูกสร้างขึ้นเลยในโปรแกรม
2. ทั้ง 4 class จงสร้าง
 - 2.1 default constructor ในการกำหนดค่าเริ่มต้นทุกค่าให้เป็นค่าว่าง หรือ 0
 - 2.2 constructor ในการกำหนดค่าทุกค่า (รวมทั้งค่าของ base class) กรณี class Person ให้ทำการจองพื้นที่ให้กับ myCar ตามจำนวนรถยนต์ที่เป็นเจ้าของด้วย
 - 2.3 destructor ให้พิมพ์ข้อมูลทั้งหมดของอีอบเจกต์ที่กำลังถูกทำลายออกทางหน้าจอ และถ้ามีการจองพื้นที่ก็ให้มีการคืนพื้นที่ที่จองด้วย
 - 2.4 ฟังก์ชัน set, get, และ show ในการกำหนดค่า ดึงค่า และ แสดงค่าทั้งหมดของแต่ละอีอบเจกต์ของแต่ละ class
3. ใน class Person จงสร้างฟังก์ชันเหล่านี้เพิ่มเติม
 - 3.1 ฟังก์ชัน setNum() ในการกำหนดค่าของ numCar และทำการจองพื้นที่ให้กับ myCar โดยให้มีขนาดเท่ากับ numCar
 - 3.2 ฟังก์ชัน setMyCar() ในการกำหนดค่าของ myCar ทั้งหมด โดยรับข้อมูลทั้งหมดจาก keyboard เช่น ถ้า numCar=3 ต้องรับค่า myCar[0], myCar[1], myCar[2] จาก keyboard นั่นคือมีการกำหนดข้อมูลของรถยนต์แต่ละคันให้กับ myCar นั่นเอง
 - 3.3 ฟังก์ชัน showMyCar() ทำการแสดงค่าที่อยู่ใน myCar ทั้งหมดออกทางจอภาพ
3. ใน main() จงสร้างอาร์เรย์ของอีอบเจกต์ของ class Person ให้มีขนาดเท่ากับ 3 และให้เขียนโปรแกรมในการกำหนดค่าทุกค่าให้กับสมาชิกแต่ละตัวในอาร์เรย์ผ่านทาง keyboard หลังจากกำหนดค่าให้ทุกอีอบเจกต์แล้วให้เขียนโปรแกรมแสดงข้อมูลของอีอบเจกต์ทุกตัวออกทางจอภาพ

หมายเหตุ: สามารถสร้าง data และ function เพิ่มเติมได้ตามความเหมาะสม