```cpp
//Sources:main.cpp
#include <iostream>
#include "sportCar.h"
#include "Truck.h"
#include "Car.h"

using namespace std;

int main()
{
    //Car mycar();
    //The class car is abstract,so it can not be instantiated.

    //sportCar e1();
    //e1.print();
    //this is error,i do not know why.

    sportCar car1("White");
    car1.print();
    std::cout<<std::endl;
    sportCar car2("Red",4,4);
    car2.print();
    std::cout<<std::endl;
    //Truck e2();
    //e2.print();
    //this also is error,i do not know why.

    Truck car3("Gray");
    car3.print();
    std::cout<<std::endl;

    Truck car4("Black",6,"Gasoline");
    car4.print();
    std::cout<<std::endl;

    Car *ptr[4];
    ptr[0]=&car1;
    ptr[1]=&car2;
    ptr[2]=&car3;
    ptr[3]=&car4;

    //C++11 for range
    for( const auto &y : ptr )
    {   // Type inference by const reference.
        // Observes in-place. Preferred when no modify is needed.
        // 参考：https://docs.microsoft.com/zh-cn/cpp/cpp/range-based-for-statement-cpp
        y->print();
    }
    /*
        以上代码等价：
        for (int i = 0; i < 5; i++)
            ptr[i]->print();
    */

    cout << endl;

    return 0;
}
```