An inheritance-based C++ application:

1. Declare and define a ***Vehicle*** class that includes data members to represent a vehicle's *model*, and *price*. The class interface includes "*set"* and "*get"* methods that provide appropriate access to the data members and an "*output"* method that prints the data information to the screen. The class interface also includes two different *constructors* (*default*, and *parameterized*) and a *destructor*.

2. Derive a ***Truck*** class from *Vehicle*. This subclass should add the following ***bool***ean data member: *dumper*. If a truck has a dumper, the value of its *dumper* attribute will be *True*, otherwise it is *False*. The interface of the subclass adds appropriate "*set*" and "*get"* methods for the *dumper*. The subclass should override the "*output"* method. You should also define the corresponding constructors and the destructor for the derived class.

3. Write a *main* function to test the *Truck* and *Vehicle* classes.

a. Instantiate one sample object from both classes and then set the corresponding information and finally output all information.

b. Create statically an array of *Truck* and another array of *Vehicle* objects and then test those objects.

c. Define a pointer to *Truck* and another pointer to *Vehicle* and then allocate the heap memory dynamically for the corresponding objects.

BONUS: You may separate the class interfaces from the class implementations, which means that you need to handle a "project" involving different files. (You may open and develop a "project" within your IDE.)