1） Create a base class with an **abstract print( )** method that is overridden in a derived class. The overridden version of the method prints the value of an **int** variable defined in the derived class. At the point of definition of this variable, give it a nonzero value. In the base-class constructor, call this method. In **main( )**, create an object of the derived type, and then call its **print( )** method. Explain the results.

2） Create an interface containing three methods, in its own **package**. Implement the interface in a different **package**

3） Create an interface, and inherit two new interfaces from that interface. Multiply inherit a third interface from the second two.

4） Create three interfaces, each with two methods. Inherit a new interface that combines the three, adding a new method. Create a class by implementing the new interface and also inheriting from a concrete class. Now write four methods, each of which takes one of the four interfaces as an argument. In **main( )**, create an object of your class and pass it to each of the methods.

5） Create a class that produces a sequence of **chars**. Adapt this class so that it can be an input to a **Scanner** object.

6） Write a class named **Outer** that contains an inner class named **Inner**. Add a method to **Outer** that returns an object of type **Inner**. In **main( )**, create and initialize a reference to an **Inner**.

7） Create a class with an inner class. In a separate class, make an instance of the inner class.

8） Create a class with a **private** field and a **private** method. Create an inner class with a method that modifies the outer-class field and calls the outer-class method. In a second outer-class method, create an object of the inner class and call its method, then show the effect on the outer-class object.

9） Determine whether an outer class has access to the **private** elements of its inner class.

10） Create an interface with at least one method, and implement that interface by defining an inner class within a method, which returns a reference to your interface.

11） Repeat previous exercise (10) using an anonymous inner class.

12） Create a class containing a nested class. In **main( )**, create an instance of the nested class.