# M6\_ShortAnswers

#### Lishuo Pan

### T1.

Q: Why is protected scope necessary?

A: If we do not use the protected to include the member variables. Then the member variables are only accessible from:

1. within the member function of the same class.
2. within the friend functions.
3. within the member function of the friend function.

It means it is not accessible for the derived classes. However, if we use the protected to include the member variables. The member variables of the higher class can be accessed by the derived class member function and all derived class's friend functions and member function of the friend function.

### T2.

Q: Discuss the wisdom, from a software engineering perspective, of using protected members.

A: If you only use the private to keep the member variable, then each time you want to use the certain member with the same name of the member variable which is created in the higher class, you have to create a new one in the current class, this will make the maintenance much harder for the developer.

However, if you use the protected, you can access the member in the higher classes. This will make the reuse of the certain member variables much more easier.

### T3.

Q: When constructing an object of a derived type, is a base class constructor always invoked? Explain.

A: Yes, if you invoke the constructor of the derived type, it will automatically invoke the higher class's constructor, and you can specify which constructor to invoke. Then the constructor of the higher classes will be executed first and then execute the lower classes constructor.

### T4.

Q: Discuss the veracity of the following statement: For an object of some derived type, the base class constructor is invoked first, then from the constructor the derived class is invoked.

A: This statement is wrong, the order of invoke of constructor is: the current class's constructor will be invoked first, then invoke the near's related higher class.

However, the executed order is: first execute highest class, then down to the current class.

### T5.

Q: Why doesn't each object that has a virtual function associated with it have its own V-table?

A: We do not have to assign each objective that has a virtual function associated with a V-table, because, the objective from the same class we can assign them with the same V-table, and then each table record the same virtual function behaviors within that class. If we assign each object that has a virtual function associated with it a V-table, then if there are many objects, this will definitely cause a waste in the memory.