Q: select the false statement regarding inheritance

1. A derived class can contain more attributes and behaviors than its base class
2. A derived class can be the base class for other derived classes
3. Some derived classes can have multiple base classes
4. Base classes are usually more specific than derived classes

A: D

※可能會考翻譯&why

Q: which of the following is not a kind of inheritance in c++?

1. Public
2. Private
3. Static
4. Protected

A: C

Q: there **is-a** relationship represents

1. Composition
2. Inheritance
3. Information hiding
4. A friend

A: B

a-> **has-a** relationship , **part-of** a relationship , **whole-part** a relationship

Q: which of the following is most likely a base class of the other three?

1. Automobile
2. Convertible
3. Minivan
4. Sedan

A: A

Q: which of the following is not a good example of a hierarchy likely to be modeled by heritance?

1. Airplanes
2. Geometric shapes
3. Animals
4. Prime numbers

A: D

Q: to declare class subClass a privately derived class of superclass

1. Class subclass : private superclass
2. Class subclass :: private superclass
3. Class subclass < private superclass
4. Class subclass inherits private superclass

A: a

Q: from most restrictive to least restrictive ,the access modifiers are:

1. Protected, private, public
2. Private, protected, public
3. Private, public, protected
4. Private, protected, public

A: d

Q: protected base class numbers cannot be accessed by:

1. Functions that are neither friends of the base class, derived-class member function … derived class
2. Friends of the base class
3. Functions that are not derived-class member function
4. Friends of derived classes

A: a

※cannot刪掉 答案是c

Q: assuming the definition

Class BasePlusCommissionEmployee : public CommissionEmployee;

Which of the following is false?

1. The colon ( : ) in the header of the class definition indicates inheritance
2. The keyword public indicates the type of inheritance
3. All the public and protected members of class BasePlusCommissionEmployee are inheritance as public and protected members, respectively ,into class CommissionEmployee
4. CommssionEmployee is the base class and BasePlusCommissionEmployee is the derived class

A: c

Q: assuming the following is the beginning of the constructor definition for class BasePlusCommissionEmployee which inherits from class Point.

**BasePlusCommissionEmployee :: BasePlusCommissionEmployee**(string first, string last, string ssn, double sales, double rate, double salary ) : CommissionEmployee (frist, last, ssn, sales, rate)

The second line:

1. Invokes the CommissionEmployee constructor with arguments
2. Causes a compiler error
3. Is unnccessary because the CommissionEmployee constructor is called automatically
4. Indicates ingeritance

A: a

Q: which of the following is not one of the disadvantages of using the “copy-and-paste” approach to duplicating code from one class into another class?

1. Errors are prone to be spread around
2. It is time consuming
3. It forces the system to store many physical copies of the code. Creating a code-maintenance nightmare
4. All of the above are disadvantages of the “copy-and-paste” approach

A: d

Q: when should base class members be declared protected?

1. When all clients should be able to access these members
2. When these members are used only by member functions of the base class
3. When these members should be available only to derived classes(and friends ), but not to other clients.
4. The protected access specified should never be used

A: c

Q: when an object of a derived class is instantiated(實例化) ,the \_\_\_\_ constructor initializes the \_\_\_\_ members

1. Base class, base class
2. Derived class ,base class
3. Base class, derived class
4. Derived class, public

A: a

Q: base class constructors and assignment operators:

1. Are not inherited by derived classes
2. Should not be called by derived class constructors and assignment operators
3. Can be inherited by derived classes ,but generally are not
4. Can be derived-class constructor and assignment operators

A: a

Q: Suppose class A inherits from base class B. what is the order in which their constructors and destructors will be called when an object of class A is instantiated and the destroyed?

1. B constructor. A constructor. A destructor. B destructor
2. B constructor. A constructor. B destructor. A destructor
3. A constructor. B constructor. A destructor. B destructor
4. A constructor. B constructor. B destructor. A destructor

A: a

Q: which forms of inheritance are is-a relationship?

1. All forms of inheritance are is-a relationship
2. Only public and private
3. Only public and protected
4. Only public

A: d

Q: theoretically, clients do not need to see the \_\_\_\_ of classes from which they derive other classes.

1. Header files
2. Source code
3. Object code
4. Interface

A: b

Q: which of the following is true about using inheritance in software engineering?

1. Common attributes and behaviors should be factored out of closely related classes and placed into a base classes from which the original classes can now inherit.
2. It is best to create a huge class library to make it easy for a client to find the most appropriate class for his or her needs
3. A class produced through inheritance should be as large as possible to fully encompass all the functionality it should offer
4. The standard c++ libraries that are shipped with c++ compilers are usually enough to accomplish anything an application might need to do

A: a