Capitulo 12

This activity contains 19 questions.

1. *Section 12.1 Introduction*

12.1 Q1: Select the false statement regarding inheritance.

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

2. *12.1 Q2: Which of the following is not a kind of inheritance in C++?*

- *public.*
- *protected.*
- *static.*
- *private.*

3. *12.1 Q3: The is-a relationship represents.*

- *Inheritance.*
- *Information Hiding.*
- *A friend.*
- *Composition.*

4. *Section 12.2 Base Classes and Derived Classes*

12.2 Q1: Which of the following is most likely a base class of the other three?

- *sedan.*
- *automobile.*
- *miniVan.*
- *convertible.*

5. *12.2 Q2: Which of the following is not a good example of a hierarchy that could be modeled by inheritance?*

- *Animals.*
- *Prime numbers.*
- *Geometric shapes.*
- *Airplanes.*

6. *12.2 Q3: To declare class subClass a privately derived class of superClass one would write:*

- *class subclass :: private superClass.*
- *class subclass < private superClass >.*
- *class subclass : private superClass.*
- *class subclass inherits private superClass.*

7. *Section 12.3 protected Members*

12.3 Q1: From most restrictive to least restrictive, the access modifiers are:

- *private, public, protected.*
- *protected, private, public.*
- *protected, public, private.*
- *private, protected, public.*

8. *12.3 Q2: protected base class members cannot be accessed by:*

- *friends of the base class.*
- *Functions that are neither friends of the base class, derived-class member functions nor friends of a derived class.*
- *friends of derived classes.*
- *Functions that are not derived-class member functions.*

9. *Section 12.4 Relationship between Base Classes and Derived Classes*

12.4 Q1: Assuming the definition,

class Circle : public Point

which of the following is false?

- *The colon (:) in the header of the class definition indicates inheritance.*
- *Point is the base class and Circle is the derived class.*
- *All the public and protected members of class Circle are inherited as public and protected members, respectively, into class Point.*
- *The keyword public indicates the type of inheritance.*

10. *12.4 Q2: Assuming the following is the beginning of the constructor definition for class Circle which inherits from class Point,*
Circle::Circle(double r, int a, int b)
: Point(a, b)
The second line:

- *Invokes the Point constructor with values a and b.*
- *Is unnecessary because the Point constructor is called automatically.*
- *Indicates inheritance.*
- *Causes a compiler error.*

11. *12.4 Q3: 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?*

- *It forces the system to store many physical copies of the code, creating a code-maintenance nightmare.*
- *It is time consuming.*
- *All of the above are disadvantages of the "copy-and-paste" approach.*
- *Errors are prone to getting spread around.*

12. *12.4 Q4: When should base class members be declared protected?*

- *When these members should be available only to derived classes (and friends), not to other clients.*
- *The protected access specified should never be used.*
- *b. When these members are only used by member functions of this base class.*
- *When all clients should be able to access these members.*

13. *Section 12.5 Constructors and Destructors in Derived Classes 12.5 Q1: When an object of a derived class is instantiated, the _____ constructor initializes the _____ members.*

- *Derived class, public.*
- *Base class, base class.*
- *Base class, derived class.*
- *Derived class, base class.*

14. *12.5 Q2: Base class constructors and assignment operators:*

- *Should not be called by derived class constructors and assignment operators.*
- *Can be inherited by derived classes, but generally are not.*
- *Are not inherited by derived classes.*
- *Can call derived-class constructors and assignment operators.*

15. *12.5 Q3: 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 then destroyed?*

- *B constructor, A constructor, B destructor, A destructor.*
- *A constructor, B constructor, A destructor, B destructor.*
- *B constructor, A constructor, A destructor, B destructor.*
- *A constructor, B constructor, B destructor, A destructor.*

16. *Section 12.6 public, protected and private Inheritance*

12.6 Q1: Which forms of inheritance are is-a relationships?

- *Only public and private.*
- *Only public.*
- *All forms of inheritance are is-a relationships.*
- *Only public and protected.*

17. *12.6 Q2: When deriving a class from a protected base class, the public members of the base class become _____ and the protected members of the base class become _____?*

- *protected, private.*
- *protected, protected.*

- *public, private.*
- *public, protected.*

18. *12.7 Q1: Theoretically, clients do not need to see the _____ of classes from which they derive other classes.*

- *Source code.*
- *Object code.*
- *Header files.*
- *Interface.*

19. *12.7 Q2: Which of the following is true about using inheritance in software engineering?*

- *Common attributes and behaviors should be factored out of closely related classes and placed into a base class from which the original classes can now inherit.*
- *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.*
- *A class produced through inheritance should be as large as possible to fully encompass all of the functionality it should offer.*
- *The standard C++ libraries that are shipped with C++ compilers are usually enough to accomplish anything an application might need to do.*

Clear Answers / Start Over Submit Answers for Grading

Answer choices in this exercise appear in a different order each time the page is loaded.