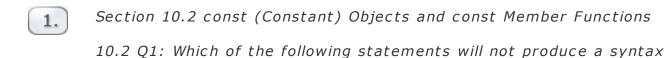
Home Exercícios de múltipla escolha (em Inglês) > Capitulo 10

Capitulo 10

This activity contains 22 questions.

error?



Defining a const member function that modifies a data member of the object.

- Declaring an object to be const.
- Declaring a constructor to be const.
- Invoking a non-const member function on a const object.
- 2. 10.2 Q2: The code fragment:

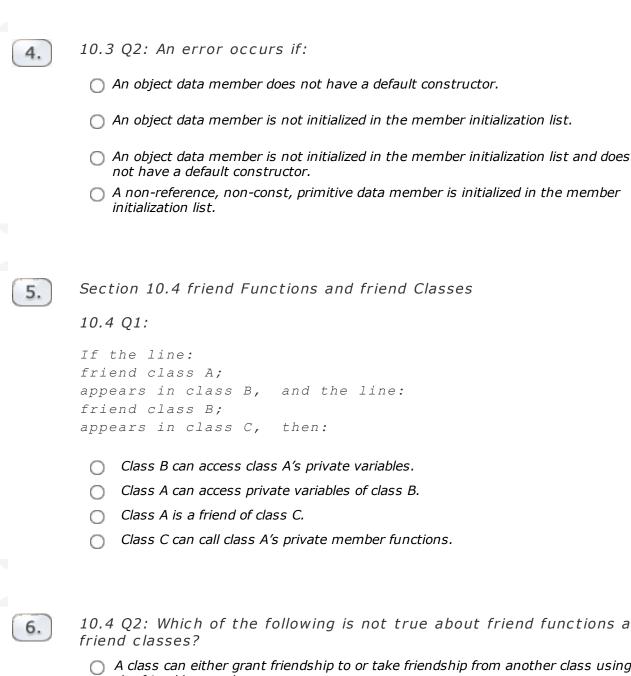
```
Increment::Increment( int c, int i )
    : increment ( i )
{
    count = c;
}
```

does not cause any compilation errors. This tells you that:

- count must be a const variable.
- count must be a non-const variable.
- increment must be a non-const variable.
- increment must be a const variable.
- 3. Section 10.3 Composition: Objects as Members of Classes

10.3 Q1: When composition (one object having another object as a member) is used:

- Member objects are constructed first, in the order they are declared in the host's class.
- Member objects are destructed last, in the order they are declared in the host's class.
- Member objects are constructed first, in the order they appear in the host constructor's initializer list.
- The host object is constructed first and then the member objects are placed into it.



10.4 Q2: Which of the following is not true about friend functions and

- A class can either grant friendship to or take friendship from another class using the friend keyword.
- A friend declaration can appear anywhere in a class definition.
- A friend of a class can access all of its private data member and member functions.
- The friendship relationship is neither symmetric nor transitive.
- Section 10.5 Using the this Pointer

10.5 Q1: For a non-constant member function of class Test, the this pointer has type:

const Test *.

	Test const *.
	○ Test * const.
	o const Test * const.
8.	10.5 Q2: Inside a function definition for a member function of an object with data element x , which of the following is not equivalent to this-> x :
	(* (& (*this))).x.
	*this.x.
	○ x.
	(*this).x.
9.	10.5 Q3: Assume that t is an object of class Test, which has member functions $a()$, $b()$, $c()$ and $d()$. If the functions $a()$, $b()$ and $c()$ all return references to an object of class Test (using the dereferenced this pointer) and function $d()$ is declared void, which of the following statements will not produce a syntax error:
	t.a().b().d();.
	○ a().b().t;.
	<pre>t.d().c();.</pre>
10.	Section 10.6 Dynamic Memory Management with Operators new and delete
	10.6 Q1: Which of the following is false about the new operator and the object for which it allocates memory?
	It automatically destroys the object after main is exited.
	It does not require the size of the object to be explicitly specified in the new expression.
	It calls the object's constructor.
	It returns a pointer.
	1. retarns a pointer.
11.	10.6 Q2: The delete operator:

O Can delete an entire array of objects declared using new.

	Must be told which destructor to call when destroying an object.	
	Can terminate the program.	
	Is called implicitly at the end of a program.	
12.	Section 10.7 static Class Members	
	10.7 Q1: If Americans are objects of the same class, which of the following attributes would most likely be represented by a static variable of that class?	
	○ Age.	
	Place of birth.	
	The President.	
	Favorite food.	
13.	10.7 Q2: static data members of a certain class:	
	Can be accessed only if an object of that class exists.	
	Have class scope.	
	Can only be changed by static member functions.	
	Cannot be changed, even by objects of the same that class.	
	Carriot be changed, even by objects of the same that class.	
14.	10.7 Q3: static member functions:	
	Can only access other static member functions and static data members.	
	Can use the this pointer.	
	Cannot be called until an object of their class is instantiated.	
	Can be declared const as well.	
	Can be declared const as well.	
15.	Section 10.8 Data Abstraction and Information Hiding	
	10.8 Q1: Which of the following is not an abstract data type?	
	O An ACCIT of a vertex	
	An ASCII character.	
	A used-defined class.	
	A for loop.	



10.8 Q2: Which of the following are true about an abstract data type?

I. Captures a data representati							7 .			
	on	1 + 7 0	t a	sen	pres	rer	data	а	ptures	. Ca

- II. Defines the operations that are allowed on its data.
- III. Replaces structured programming.

\cap	I and	III.

- II and III.
- I and II.
- I, II and III.



Section 10.8.1 Example: Array Abstract Data Type

10.8.1 Q1: Which of the following capabilities do "raw" C++ arrays not provide?

- "Raw" arrays do not provide any of the above capabilities.
- Array comparison.
- Subscript range checking.
- Opposition of the property of the property

18.

Section 10.8.2 Example: String Abstract Data Type

10.8.2 Q1: Instead of including a string data type among C++'s built-in data types, C++: a. Was designed to include mechanisms for creating and implementing string abstract data types through classes.

- Was designed to include mechanisms for creating and implementing string abstract data types through classes.
- Forces the programmer to make do with char array strings.
- Chose to ignore the need for a string data type.
- None of the above.



Section 10.8.3 Example: Queue Abstract Data Type

10.8.3 Q1: The numbers 3, 2, 5, 7 are enqueued in a queue in that order, then three numbers are dequeued, and finally 3, 7, 9, 4 are enqueued in that order. What is the first number in the queue (the next number to be dequeued)?

	O 4.
	○ 3.
	O 9.
	O 7.
20.	Section 10.9 Container Classes and Iterators
	10.9 Q1: Which of the following is not a type of container (collection) class?
	Linked lists.
	Stacks.
	ofloats.
	O Arrays.
21.	Section 10.10 Proxy Classes
	10.10 Q1: Proxy classes are best described as an example of:
	Information hiding.
	Utility functions.
	Structured programming.
	Object-oriented programming (as used in the text).
22.	10.10 Q2: In addition to hiding the implementation details that the ordinary method of "separating implementation from interface" would hide, using a proxy class also hides:
	The definition of constructors and the destructor.
	The definition of access functions.
	The names of private data members.
	The definition of inline functions.
	Clear Answers / Start Over

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



Copyright © 1995 - 2010 Pearson Education . All rights reserved.

<u>Legal Notice</u> | <u>Privacy Policy</u> | <u>Permissions</u>