Results

100%

8

Out of 8 points

This assessment has unlimited attempts. Take Now

Your Answers:

1 1 / 1 point

What is the purpose of an abstract data type (ADT)?

- To implement a virtual function that can be overridden by a subclass.
- ✓ To encapsulate a data structure and its associated operations
- To create a hierarchy of derived classes that share a common base class.
- To define a class with private data members and public member functions.

2 1 / 1 point

Which of the following are examples of ADTs available from the C++ libraries?

- int
- ✓ priority_queue
- ✓ vector
- ✓ string

3 1 / 1 point

Which of the following is an example of information hiding in an ADT?

- A constant data member in a class.
- ✓ A private data member in a class.
- A static data member in a class.
- A public data member in a class.

4 1 / 1 point

What is an advantage of using an ADT in C++?

- It makes the code more efficient.
- ✓ It makes the code more maintainable.
- ✓ It makes the code easier to debug.
- It reduces the memory needed.

5 1 / 1 point

Which of the following parts of a class should be declared as private to achieve information hiding for an ADT in C++?

- None of the data members nor member functions should be private. They should be public so they can be accessed.
- ✓ Functions that are not part of the interface (helper functions)
- All class members.
- ✓ All data members.
- All member functions.

6 1 / 1 point

Every data member should always have a public getter and setter function.

- True

- ✓ False

7 1 / 1 point

Which of the following scenarios best illustrates a violation of the Single Responsibility Principle?

- A class that performs encryption and also manages network connections.
- A class that generates random numbers and also calculates the sum of an array.
- ✓ A class that validates user input and also formats data for display.
- A class that represents a database table and also performs data validation.

8 1 / 1 point

Which of the following scenarios best illustrates a violation of the Liskov Substitution Principle?

- A subclass that inherits from a base class and adds new methods.
- ✓ A subclass that inherits from a base class and changes the behavior of a method in a way that is not compatible with the base class.
- A subclass that inherits from a base class and has additional properties.
- A subclass that inherits from a base class and overrides a method with the same behavior.