Capitulo 16

This activity contains 24 questions.

1. *Section 16.1 Introduction*

16.1 Q1: Exception handling may allow a program to:

- ○ *Terminate in a controlled manner.*
- ○ *Be more robust and fault-tolerant.*
- ○ *Continue executing as if no problem was encountered.*
- ○ *All of the above.*

2. *Section 16.2 Exception-Handling Overview*

16.2 Q1: Which of the following is not one of the disadvantages of not using exception-handling to deal with errors?

- ○ *Programmers may delay writing error-processing code or sometimes forget to include it.*
- ○ *All of the above are disadvantages of not using exception-handling to deal with errors.*
- ○ *Frequent tests for infrequently occurring errors can degrade a program's performance.*
- ○ *Intermixing program logic with error-handling logic can make the program more difficult to read, modify, maintain and debug.*

3. *Section 16.3 Example: Handling an Attempt to Divide by Zero*

16.3 Q1: The correct order in which an exception is detected and handled is:

- ○ *catch, throw, try.*
- ○ *throw, catch, try.*
- ○ *try, throw, catch.*
- ○ *try, catch, throw.*

4. *16.3 Q2: Once an exception is thrown, when can control return to the throw point?*

- ☐ *Never.*
- ☐ *Immediately after the exception is thrown.*

- ☐ *Once the stack unwinding process is completed.*
- ☐ *Only after the exception is caught.*

5. *16.3 Q3: The try block cannot:*

- ◯ *Enclose the code that may throw the exception.*
- ◯ *Have exceptions explicitly or implicitly thrown in the try block itself.*
- ◯ *Test enclosing try blocks for additional catch statements if this try block's catch statements can't match the exception being thrown.*
- ◯ *Enclose its own catch blocks.*

6. *16.3 Q4: catch blocks are not required to contain:*

- ◯ *Some form of parameter type indication.*
- ◯ *Parentheses ( ).*
- ◯ *A parameter name.*
- ◯ *Braces { }.*

7. *16.3 Q5: An exception:*

- ◯ *Will terminate the block where the exception occurred unless a catch command stops it.*
- ◯ *Terminates program execution.*
- ◯ *Will not terminate a block unless explicitly instructed to do so.*
- ◯ *Terminates the block where the exception occurred.*

8. *Section 16.4 When to Use Exception Handling*

16.4 Q1: Exception handling should not be used:

- ◯ *To deal with errors for components that will be widely used in other applications, such as classes and libraries.*
- ◯ *To deal with errors that do not arise very often.*
- ◯ *To make error handling uniform on large projects.*

- *As an alternative for program control.*

9. *Section 16.5 Rethrowing an Exception*

16.5 Q1: To rethrow an exception, the exception handler must:

- *Use the throw; statement.*
- *Use the throw command with the same parameters as the original exception.*
- *d. Not have attempted to process that exception at all.*
- *Return a reference to whatever caused the original exception.*

10. *16.5 Q2: Select the false statement. A rethrown exception:*

- *Can be processed by exception handlers following the enclosing try block.*
- *Must have been fully processed at the time it was rethrown.*
- *Is detected by the next enclosing try block.*
- *Is the immediate result of a throw; command.*

11. *Section 16.6 Exception Specifications*

16.6 Q1: The proper syntax for a throw list is:

- *a.int g(double h)*
  throw (a, b, c).
- *int g(double h)*
  throw (a),
  throw (b),
  throw (c).
- *int g(double h)*
  throw (a)
  throw (b)
  throw (c).
- *int g(double h)*
  throw (a, b, c).

12. *16.6 Q2: Placing throw() after a function's parameter list:*

- *Guarantees that all exceptions can be thrown in this function.*
- *d. Indicates that the compiler will issue an error if the function contains a throw expression.*
- *Indicates that throwing an exception in this function would call unexpected.*

- *Guarantees that only programmer-defined exceptions can be thrown in this function.*

13. *Section 16.7 Processing Unexpected Exceptions*

16.7 Q1: Select the false statement. The functions set_terminate and set_unexpected:

- *Return pointers to the last function called by terminate and unexpected, respectively.*
- *Take as arguments pointers to void functions with no arguments.*
- *Each return 0 the first time they are called.*
- *d. Have their prototypes in header file .*

14. *16.7 Q2: Which of the following is not a case in which function terminate is called?*

- *When an attempt is made to rethrow an exception when there is no exception currently being handled.*
- *When the abort function is called before any call to function set_abort.*
- *When a destructor attempts to throw an exception during stack unwinding.*
- *When the exception mechanism cannot find a matching catch for a thrown exception.*

15. *Section 16.8 Stack Unwinding*

16.8 Q1: The purpose of stack unwinding is to:

- *b. Improve catch blocks by allowing them to handle multiple exceptions.*
- *Return control to the function that created the exception.*
- *Attempt to catch exceptions that are not caught in their scope.*
- *Aid the terminate command in shutting down the program.*

16. *Section 16.9 Constructors, Destructors and Exception Handling*

16.9 Q1: Select the false statement. If an exception is thrown from a constructor:

- *The object being constructed will not be constructed.*
- *The exception can contain the error information that the constructor would not be*

able to return in the normal manner.

- *For an object with member objects, and whose outer object has not been constructed, the destructor is called for the member objects.*
- *For an array, destructors for all array elements are called, even if those array elements have not yet been constructed.*

17. *Section 16.10 Exceptions and Inheritance*

16.10 Q1: An advantage of using inheritance with exceptions is:

- *Allowing catch statements to be imported into classes.*
- *The ability to catch related errors easily.*
- *The ability to explicitly test for derived class objects individually.*
- *The simplification of destructor calls for objects.*

18. *Section 16.11 Processing new Failures*

16.11 Q1: Select the false statement. Depending on the compiler:

- *A failed new command can throw an exception if the header file has been included.*
- *A failed new command can throw a bad_alloc exception.*
- *A failed new command can automatically be caught at compile time.*
- *A failed new command can return a 0.*

19. *16.11 Q2: Select the false statement. The new operator:*

- *Can indicate failure differently on different compilers.*
- *Can attempt to allocate as much memory as the programmer requests.*
- *Throws a bad_alloc exception regardless of what function is registered with set_new_handler.*
- *Returns a pointer to a location in memory.*

20. *Section 16.12 Class auto_ptr and Dynamic Memory Allocation*

16.12 Q1: If dynamic memory has been allocated for an object and an exception occurs, then:

Multiple pointers to memory could be created.

- *The catch block will not work properly.*
- *The object's constructor will cause another exception.*
- *A memory leak could result.*

21. *Section 16.13 Standard Library Exception Hierarchy*

16.13 Q1: Select the false statement regarding exceptions.

- *Several classes derive from class exception.*
- *The what function can be overridden in each class derived from exception.*
- *All exception classes are accessible via .*
- *The C++ standard has a hierarchy of exception classes.*

22. *16.13 Q2: Which class indicates that an error occurred in which an arithmetic result was larger than the largest number that can be stored in the computer?*

- *invalid_argument.*
- *overflow_error.*
- *bad_exception.*
- *out_of_range.*

23. *Section 16.14 Other Error-Handling Techniques*

16.14 Q1: Which of the following is not an error-handling technique?

- *ifndef.*
- *set_new_handler.*
- *exit.*
- *longjump.*

24. *16.14 Q2: Both "ignoring the exception" and "aborting the program" are error-handling techniques that:*

- *Allow program execution to proceed as if no error had occurred.*
- *Should not be used for mission-critical applications.*
- *Always result in a resource leak.*
- *Cannot be used if the error is fatal.*

Clear Answers / Start Over Submit Answers for Grading

Some questions in this exercise may have more than one correct answer. To answer such questions correctly, you must select all the correct answers. Also note that answer choices in this exercise appear in a different order each time the page is loaded.