1. D. A try block must include either a catch or finally block, or both. The think()

method declares a try block but neither additional block. For this reason, the code

does not compile, and Option D is the correct answer. The rest of the lines compile

without issue, including k1.

2. B. The correct order of blocks is try, catch, and finally, making Option B the correct

answer.

3. D. Option D is the correct model. The class RuntimeException extends Exception, and

both Exception and Error extend Throwable. Finally, like all Java classes, they all

inherit from Object. Notice that Error does not extend Exception, even though we

often refer to these generally as exceptions.

4. A. While Exception and RuntimeException are commonly caught in Java applications,

it is recommended Error not be caught. An Error often indicates a failure of the JVM

which cannot be recovered from. For this reason, Option A is correct, and Options C

and D are incorrect. Option B is not a class defined in the Java API; therefore, it is also

incorrect.

5. D. The application does not compile because score is defined only within the try

block. The other three places it is referenced, in the catch block, in the finally block,

and outside the try-catch-finally block at the end, are not in scope for this variable and

each does not compile. Therefore, the correct answer is Option D.

6. B. ClassCastException, ArrayIndexOutOfBoundsException, and

IllegalArgumentException are unchecked exceptions and can be thrown at any time.

IOException is a checked exception that must be handled or declared when used,

making Option B the correct answer.

7. A. The throws keyword is used in method declarations, while the throw keyword is

used to throw an exception to the surrounding process, making Option A the correct

answer. The catch keyword is used to handle exceptions, not to create them or in the

declaration of a method.

8. B. IOException is a subclass of Exception, so it must appear first in any related catch

blocks. If Exception was to appear before IOException, then the IOException block

would be considered unreachable code because any thrown IOException is already

handled by the Exception catch block. For this reason, Option B is correct.

9. C. The application first enters the try block and outputs A. It then throws a

RuntimeException, but the exception is not caught by the catch block since

RuntimeException is not a subclass of ArrayIndexOutOfBoundsException (it is a

superclass). Next, the finally block is called and C is output. Finally, the

RuntimeException is thrown by the main() method and a stack trace is printed. For

these reasons, Option C is correct.

10. C. The application does not compile, so Option D is incorrect. The openDrawbridge()

method compiles without issue, so Options A and B are incorrect. The issue here is

how the openDrawbridge() method is called from within the main() method on line p3.

The openDrawbridge() method declares the checked exception, Exception, but the

main() method from which it is called does not handle or declare the exception. In

order for this code to compile, the main() method would have to have a try-catch

statement around line p3 that properly handles the checked exception, or the main()

would have to be updated to declare a compatible checked exception. For these

reasons, line p3 does not compile, and Option C is the correct answer.

11. B. NullPointerException and ArithmeticException both extend RuntimeException,

which are unchecked exceptions and not required to be handled or declared in the

method in which they are thrown. On the other hand, Exception is a checked

exception and must be handled or declared by the method in which it is thrown.

Therefore, Option B is the correct answer.

12. A. The code compiles and runs without issues, so Options C and D are incorrect. The

try block throws a ClassCastException. Since ClassCastException is not a subclass of

ArrayIndexOutOfBoundsException, the first catch block is skipped. For the second

catch block, ClassCastException is a subclass of Throwable, so that block is executed.

Afterward, the finally block is executed and then control returns to the main()

method with no exception being thrown. The result is that 1345 is printed, making

Option A the correct answer.

13. C. A finally block can throw an exception, in which case not every line of the finally

block would be executed. For this reason, Options A and D are incorrect. Option B is

also incorrect The finally block is called regardless of whether or not the related

catch block is executed. Option C is the correct answer. Unlike an if-then statement,

which can take a single statement, a finally statement requires brackets {}.

14. C. The code does not compile because the catch blocks are used in the wrong order.

Since IOException is a superclass of FileNotFoundException, the

FileNotFoundException is considered unreachable code. For this reason, the code does

not compile, and Option C is correct.

15. C. A try statement requires a catch or a finally block. Without one of them, the code

will not compile; therefore, Option D is incorrect. A try statement can also be used

with both a catch and finally block, making Option C the correct answer. Note that

finalize is not a keyword, but a method inherited from java.lang.Object.

16. B. Option A is a true statement about exceptions and when they are often applied.

Option B is the false statement and the correct answer. An application that throws an

exception can choose to handle the exception and avoid termination. Option C is also a

true statement. For example, a NullPointerException can be avoided on a null object

by testing whether or not the object is null before attempting to use it. Option D is

also a correct statement. Attempting to recover from unexpected problems is an

important aspect of proper exception handling.

17. D. The code does not compile because the catch block uses parentheses () instead of

brackets {}, making Option D the correct answer. Note that Boat does not extend

Transport, so while the override on line j1 appears to be invalid since Exception is a

broader checked exception than CapsizedException, that code compiles without issue.

If the catch block was fixed, the code would output 4, making Option A the correct

answer.

18. B. Overridden methods cannot throw new or broader checked exceptions than the one

they inherit. Since Exception is a broader checked exception than PrintException,

Option B is not allowed and is the correct choice. Alternatively, declaring narrower or

the same checked exceptions or removing them entirely is allowed, making Options A

and C incorrect. Since Option B is correct, Option D is incorrect.

19. D. All three of those classes belong to the java.lang package, so Option C seems like

the correct answer. The Java compiler, though, includes java.lang by default, so no

import statement is actually required to use those three classes, making Option D the

correct answer.

20. C. The code does not compile because the catch block is missing a variable type and

name, such as catch (Exception e) . Therefore, Option C is the correct answer. Both

implementations of getSymbol() compile without issue, including the overridden

method. A subclass can swallow a checked exception for a method declared in a parent

class; it just cannot declare any new or broader checked exceptions.