Chapter 11 – Exceptions

Answers

1. A,D,E

   The program will go into the catch block on line 11 that accepts a "FirstException" as a parameter, and the word "First" will be printed. Then the program will go into the finally block on line 19 and the word "finally" will be printed. Since the exception was caught, the code will then continue and the text "keep on going" on line 23 will be printed.

2. B

   This "throw" statement demonstrates the best way to throw an exception. Since an exception builds its descriptive information from the point that it was created, it is best to create the exception and throw it in the same statement, so no information will be missing.

3. D

   A "Berror" extends from "Aerror", and that class extends from "Error", which extends directly from "Throwable". There is no catch block created for either of these class types, so this error is not caught. The program runs the code in the finally block in any case, so the text "doing finally part" is printed. Then since the error was not caught the method ends here, and the text on line 23 is not printed.

4. D,E,F

   If no errors occur then the text on line 5, "Success" is printed. Then the text in the finally block is printed, "doing finally part". And then the program continues and the text on line 23 is printed "Carrying on".

5. A,C

   An overriding method does not have to throw the exceptions that its original method throws, but if it does throw one or more exceptions, they need to be either the same type or sub class types of the

exceptions for the original method. An overriding method cannot have a more restrictive accessibility than its original method.

6. D

   In this case a “BambaException” extends from “Exception”, so if there were catch blocks for either one of these exception types, the exception would have been caught. Since this is not the case the program jumps to the finally block and the text on line 21 “finally” is printed, and then the method is exited since an exception was thrown that was not caught.

7. A,C,E

   The method in answer A has a different parameter list and a different return type, so this is not a case of overriding and is just a newly defined method, so it is OK. The method in answer C is defined exactly as the original method to override, so this is an acceptable form of an overridden method. The method in answer E is also an acceptable overriding method, since it is defined similarly to the original method, its only difference being that the exception that it throws is a sub class of the exception thrown in the original method, which is acceptable.

   Answer B is not correct since it is trying to override the original method, but its only change is that its return type is different, and this is not an acceptable override.

8. B,F

   Answer B is correct since the name of the parameter that is accepted by the “getParameter” method is not case sensitive, so the text returned from the this method is “Israel2000”. Then this text is converted to uppercase, to be “ISRAEL2000”, and then it is placed into a new String object who’s reference is stored in the “str” variable.

   Answer E is correct since although the “toUpperCase()” method will throw a RunTime error if the String value that it is working on is null, it is not required that a RunTime method be caught. The code will

compile without the try and catch block here. The code should just be written properly to handle this situation if it were to occur.

9. A,B,C
   If the stop method is run on a thread before the finally block is run, then it will never be run. Or if an exception is thrown in the finally block and is not caught properly then this block will exit. The System.exit() method stops the whole JVM, so for sure the finally blcok code would not run in this case, as well as all other code.

   If an exception is caught properly before the finally block, or if no exception was thrown, the finally block executes.

10. D
    Since a “BambaException” is a RunTime exception, and there are no catch blocks that are fitting to catch it, the code in the finally block runs, printing the word “finally” and then the method exits, so the text on line 23 is not printed. Usually RunTime exceptions are not placed into try and catch blocks, since the code should be written properly so that they do not occur at all.

11. A
    An applet cannot read files on the local computer that it is running on and it cannot write files to the local computer as well.

12. C, D, E
    Since there is no catch block to accept an IOException, the compiler looks for and finds a catch block that accepts its sub class as an exception, which is type “Exception”. This code then executes and the text “General” on line 17 is written. Then the text in the finally block on line 21 is written “finally”. Since the exception was caught properly, the program then continues and the text on line 23, “keep on going” is printed.
13. D
    Since there is no catch block to accept an IOException or its super class Exception the program goes straight to the finally block, prints

the text “finally” on line 17, and then exits the method.

14. D, E, F

    Since FourthException is a sub class of SecondException, the catch block that accepts a SecondException reference as a parameter is executed and the text “Two” on line 9 is printed. Then the text in the finally block on line 17 is printed, “finally”. Since the exception was properly caught, the program then continues and the text on line 19 “keep on going” is printed as well.

15. A, D, E

    Since there is a catch block that accepts a FirstException as a parameter, and it occurs before any catch blocks that may accept a super class of FirstClass, the program executes the code on line 13, and prints the text “First”. Then the text in the finally block on line 17 is printed, “finally”. Since the exception was properly caught, the program then continues and the text on line 19 “keep on going” is printed as well.

16. A, D, E

    Since an SQLException extends from the class Exception, the catch block that starts on line 11, which accepts an Exception type as a parameter is executed, and the text on line 13 “First” is printed. After that the text in the finally block on line 17 is printed, “finally”. Since the exception was properly caught, the program then continues and the text on line 19 “keep on going” is printed as well.

17 A, B, C, D, E

    Answers A, B and E are all examples of new methods, and are not examples of overridden methods, so therefore they are fine.
    Answers C and D are overriding the original method. Answer C is identical to the original method, and therefore is acceptable, and answer D throws an exception that is a subclass of the exception thrown ion the original method, so this is acceptable as well.

18. A

These statements about exceptions are true.

19. A

    A finally block is usually placed after try and catch blocks. The code in a finally block runs if an exception is thrown and caught, thrown and not caught, or not thrown at all.

20. A

    A RunTime exception should not occur in a program, if it does then it means that there is a bug in the program. Therefore the compiler does not require these exceptions to be caught since they should not occur anyway. All other types of Exception objects need to be handled by either try and catch blocks or by being thrown, otherwise a compilation error will occur.

21. A

    These statements are true.

22. A

    This code contains a “while” loop that runs endlessly since it has “true” within the parameters, and also because it has a try and catch block to catch an “OutOfMemoryError” error, which when entered cleans up so that more memory is provided.

23. A

    This statement is true.