

OCA / OCP Java SE 8 Programmer Practice Tests

PREV
Chapter 7 Working with Inheritance

↩ AA ⌵ 🔍

NEXT
Chapter 9 Working with Selected Classes from the Java API

Chapter 8

Handling Exceptions

THE OCA EXAM TOPICS COVERED IN THIS PRACTICE TEST INCLUDE THE FOLLOWING:

- ✓ **Handling Exceptions**
- Differentiate among checked exceptions, unchecked exceptions, and Errors
- Create a try-catch block and determine how exceptions alter normal program flow
- Describe the advantages of Exception handling
- Create and invoke a method that throws an exception
- Recognize common exception classes (such as NullPointerException, ArithmeticException, ArrayIndexOutOfBoundsException, ClassCastException)

1. What is the result of compiling and executing the following application?

```
package mind;
public class Remember {
    public static void think() throws Exception { // k1
        try {
            throw new Exception();
        }
    }
    public static void main(String... ideas) throws Exception {
        think();
    }
}
```

1. The code compiles and runs without printing anything.

2. The code compiles but a stack trace is printed at runtime.

3. The code does not compile because of line k1.

4. The code does not compile for another reason.

2. Choose the answer that lists the keywords in the order that they would be used together.

1. catch, try, finally

2. try, catch, finally

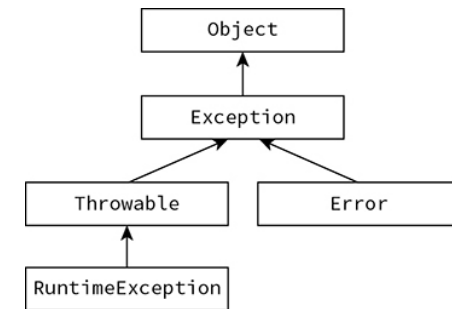
3. finally, catch, try

4. try, finally, catch

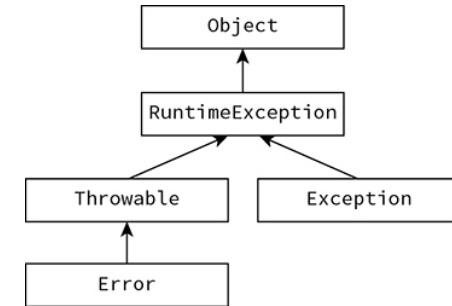
3. Which of the following diagrams of java.lang classes shows the inheritance model properly?

1.

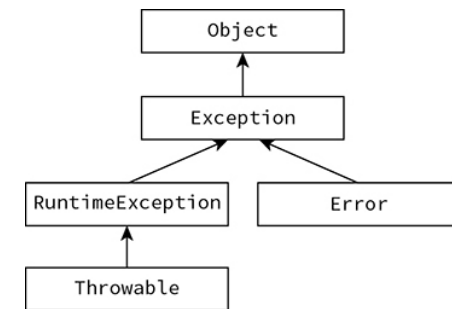
You have 2 days left in your trial, Gtucker716. Subscribe today. [See pricing options.](#)



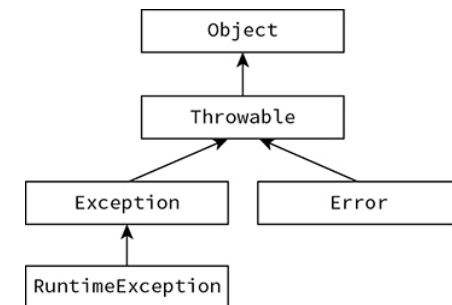
2.



3.



4.



4. Which of the following Throwable types is it recommended not to catch in a Java application?

1. Error
2. CheckedException
3. Exception
4. RuntimeException

5. What is the output of the following application?

```

package game;
public class Baseball {
    public static void main(String... teams) {
        try {
            int score = 1;
            System.out.print(score++);
        }
    }
}
  
```

```

        } catch (Throwable t) {
            System.out.print(score++);
        } finally {
            System.out.print(score++);
        }
        System.out.print(score++);
    }
}

```

1. 123
2. 124
3. 12
4. None of the above

6. Which of the following is a checked exception?

1. ClassCastException
2. IOException
3. ArrayIndexOutOfBoundsException
4. IllegalArgumentException

7. Fill in the blanks: The _____ keyword is used in method declarations, while the _____ keyword is used to throw an exception to the surrounding process.

1. throws, throw
2. catch, throw
3. throw, throws
4. throws, catch

8. If a try statement has catch blocks for both Exception and IOException, then which of the following statements is correct?

1. The catch block for Exception must appear before the catch block for IOException.
2. The catch block for IOException must appear before the catch block for Exception.
3. The catch blocks for these two exception types can be declared in any order.
4. A try statement cannot be declared with these two catch block types because they are incompatible.

9. What is the output of the following application?

```

package game;
public class Football {
    public static void main(String officials[]) {
        try {
            System.out.print('A');
            throw new RuntimeException("Out of bounds!");
        } catch (ArrayIndexOutOfBoundsException aioobe) {
            System.out.print('B');
            throw t;
        } finally {
            System.out.print('C');
        }
    }
}

```

1. ABC
2. ABC, followed by a stack trace for a RuntimeException
3. AC, followed by a stack trace for a RuntimeException
4. None of the above

10. What is the result of compiling and running the following application?

```

package castles;
public class Fortress {
    public void openDrawbridge() throws Exception { // p1
        try {
            throw new Exception("Circle");
        } catch (Exception e) {
            System.out.print("Opening!");
        } finally {
            System.out.print("Walls"); // p2
        }
    }
}

```

```

    }
    public static void main(String[] moat) {
        new Fortress().openDrawbridge(); // p3
    }
}

```

1. The code does not compile because of line p1.
 2. The code does not compile because of line p2.
 3. The code does not compile because of line p3.
 4. The code compiles, but a stack trace is printed at runtime.
11. Which of the following exception types must be handled or declared by the method in which they are thrown?

1. NullPointerException
2. Exception
3. RuntimeException
4. ArithmeticException

12. What is the output of the following application?

```

package game;
public class BasketBall {
    public static void main(String[] dribble) {
        try {
            System.out.print(1);
            throw new ClassCastException();
        } catch (ArrayIndexOutOfBoundsException ex) {
            System.out.print(2);
        } catch (Throwable ex) {
            System.out.print(3);
        } finally {
            System.out.print(4);
        }
        System.out.print(5);
    }
}

```

1. 1345
2. 1235
3. The code does not compile.
4. The code compiles but throws an exception at runtime.

13. Which of the following statements about a finally block is true?

1. Every line of the finally block is guaranteed to be executed.
2. The finally block is executed only if the related catch block is also executed.
3. The finally statement requires brackets {}.
4. The finally block cannot throw an exception.

14. Given that FileNotFoundException is a subclass of IOException, what is the output of the following application?

```

package office;
import java.io.*;
public class Printer {
    public void print() {
        try {
            throw new FileNotFoundException();
        } catch (IOException exception) {
            System.out.print("Z");
        } catch (FileNotFoundException enfe) {
            System.out.print("X");
        } finally {
            System.out.print("Y");
        }
    }
    public static void main(String... ink) {
        new Printer().print();
    }
}

```

1. XY
2. ZY
3. The code does not compile.
4. The code compiles but a stack trace is printed at runtime.

15. Which keywords are required with a try statement?

1. catch
2. finalize
3. finally
1. I only
2. II only
3. I or III, or both
4. None of these statements are required with a try statement.

16. Which statement about the role of exceptions in Java is incorrect?

1. Exceptions are often used when things “go wrong” or deviate from the expected path.
2. An application that throws an exception will terminate.
3. Some exceptions can be avoided programmatically.
4. An application that can properly handle its exception may recover from unexpected problems.

17. What is the output of the following application?

```
package harbor;
class CapsizedException extends Exception {}
class Transport {
    public int travel() throws CapsizedException { return 2; };
}
public class Boat {
    public int travel() throws Exception { return 4; }; // j1
    public static void main(String... distance) throws Exception{
        try {
            System.out.print(new Boat().travel());
        } catch (Exception e) {
            System.out.print(8);
        }
    }
}
```

1. 4
2. 8
3. The code does not compile due to line j1.
4. The code does not compile for another reason.

18. Which of following method signatures would not be allowed in a class implementing the Printer interface?

```
class PrintException extends Exception {}
class PaperPrintException extends PrintException {}

public interface Printer {
    abstract int printData() throws PrintException;
}
```

1. public int printData() throws PaperPrintException
2. public int printData() throws Exception
3. public int printData()
4. None of the above

19. Which import statement is required to be declared in order to use the Exception, RuntimeException, and Throwable classes in an application?

1. import java.exception.*;
2. import java.util.exception.*;
3. import java.lang.*;
4. None of the above

20. Which statement about the following classes is correct?

```
class GasException extends Exception {}
class Element {
```

```

        public int getSymbol() throws GasException { return -1; } // g1
    }
    public class Oxygen extends Element {
        public int getSymbol() { return 8; } // g2
        public void printData() {
            try {
                System.out.print(getSymbol());
            } catch { // g3
                System.out.print("Unable to read data");
            }
        }
    }
}

```

-
1. The code does not compile because of line g1.
 2. The code does not compile because of line g2.
 3. The code does not compile because of line g3.
 4. None of the above

21. Fill in the blanks: A program must handle or declare _____ but should never handle _____.

1. java.lang.Error, unchecked exceptions
2. checked exceptions, java.lang.Error
3. java.lang.Throwable, java.lang.Error
4. unchecked exceptions, java.lang.Exception

22. What is the result of compiling and running the following application?

```

package castles;
class CastleUnderSiegeException extends Exception {}
class KnightAttackingException extends CastleUnderSiegeException {}
public class Citadel {
    public void openDrawbridge() throws RuntimeException { // q1
        try {
            throw new KnightAttackingException();
        } catch (Exception e) {
            throw new ClassCastException();
        } finally {
            throw new CastleUnderSiegeException(); // q2
        }
    }
    public static void main(String[] moat) {
        new Citadel().openDrawbridge(); // q3
    }
}

```

-
1. The code does not compile because of line q1.
 2. The code does not compile because of line q2.
 3. The code does not compile because of line q3.
 4. The code compiles, but a stack trace is printed at runtime.

23. If an exception matches two or more catch blocks, which catch block is executed?

1. The first one that matches is executed.
2. The last one that matches is executed.
3. All matched blocks are executed.
4. It is not possible to write code like this.

24. What is the output of the following application?

```

package system;
public class Computer {
    public void compute() throws Exception {
        throw new RuntimeException("Error processing request");
    }
    public static void main(String[] bits) {
        try {
            new Computer().compute();
            System.out.print("Ping");
        } catch (NullPointerException e) {
            System.out.print("Pong");
            throw e;
        }
    }
}

```

1. Ping
2. Pong
3. The code does not compile.
4. The code compiles but throws an exception at runtime.

25. In the following application, the value of `list` has been omitted. Assuming the code compiles without issue, which one of the following is not a possible output of executing this class?

```
package checkboard;

public class Attendance {
    private Boolean[] list = // value omitted
    public int printTodaysCount() {
        int count=0;
        for(int i=0; i<10; i++) {
            if(list[i]) ++count;
        }
        return count;
    }
    public static void main(String[] roster) {
        new Attendance().printTodaysCount();
    }
}
```

1. A stack trace for `NullPointerException` is printed.
2. A stack trace for `ArrayIndexOutOfBoundsException` is printed.
3. A stack trace for `ClassCastException` is printed.
4. None of the above

26. Fill in the blanks: A _____ occurs when a program recurses too deeply into an infinite loop, while a(n) _____ occurs when a reference to a nonexistent object is acted upon.

1. `NoClassDefFoundError`, `StackOverflowError`
2. `StackOverflowError`, `NullPointerException`
3. `ClassCastException`, `IllegalArgumentException`
4. `StackOverflowError`, `IllegalArgumentException`

27. Which of the following is not a reason to add checked exceptions to a method signature?

1. To force a caller to handle or declare its exceptions
2. To notify the caller of potential types of problems
3. To ensure that exceptions never cause the application to terminate
4. To give the caller a chance to recover from a problem

28. What is the output of the following application?

```
package peculiar;
public class Stranger {
    public static String getFullName(String firstName, String lastName) {
        try {
            return firstName.toString() + " " + lastName.toString();
        } finally {
            System.out.print("Finished!");
        } catch (NullPointerException npe) {
            System.out.print("Problem?");
        }
        return null;
    }
    public static void main(String[] things) {
        System.out.print(getFullName("Joyce", "Hopper"));
    }
}
```

1. Joyce Hopper
2. Finished!Joyce Hopper
3. Problem?Finished!null
4. None of the above

29. Fill in the blanks: A try statement has _____ finally block(s) and _____ catch blocks.

1. zero or one, zero or more
2. one, one or more
3. zero or one, zero or one
4. one or more, zero or one

30. What is the output of the following application?

```
package pond;
abstract class Duck {
    protected int count;
    public abstract int getDuckies();
}
public class Ducklings extends Duck {
    private int age;
    public Ducklings(int age) { this.age = age; }
    public int getDuckies() { return this.age/count; }
    public static void main(String[] pondInfo) {
        Duck itQuacks = new Ducklings(5);
        System.out.print(itQuacks.getDuckies());
    }
}
```

1. 0
2. 5
3. The code does not compile.
4. The code compiles but throws an exception at runtime.

31. Given a try statement, if both the catch block and the finally block each throw an exception, what does the caller see?

1. The exception from the catch block
2. The exception from the finally block
3. Both the exception from the catch block and the exception from the finally block
4. None of the above

32. What is the output of the following application?

```
package zoo;
class BigCat {
    void roar(int level) throw RuntimeException { // m1
        if(level<3) throw new IllegalArgumentException("Incomplete");
        System.out.print("Roar!");
    }
}
public class Lion extends BigCat {
    public void roar() { // m2
        System.out.print("Roar!!!");
    }

    public static void main(String[] cubs) {
        final BigCat kitty = new Lion(); // m3
        kitty.roar(2);
    }
}
```

1. The code does not compile because of line m1.
2. The code does not compile because of line m2.
3. The code does not compile because of line m3.
4. The code compiles but a stack trace is printed at runtime.

33. Given the following code snippet, which specific exception will be thrown?

```
final Object exception = new Exception();
final Exception data = (RuntimeException)exception;
System.out.print(data);
```

1. ClassCastException
2. RuntimeException
3. NullPointerException
4. None of the above

34. Which of the following classes will handle all types in a catch block?

1. Exception
2. Error
3. Throwable
4. RuntimeException

35. In the following application, the values of `street` and `city` have been omitted. Which one of the following is a possible output of executing this class?

1. 350 5th Ave - New York
2. Posted:350 5th Ave - New York

```
package registration;
public class Address {
    public String getAddress(String street, String city) {
        try {
            return street.toString() + " : " + city.toString();
        } finally {
            System.out.print("Posted:");
        }
    }

    public static void main(String[] form) {
        String street = // value omitted
        String city = // value omitted
        System.out.print(new Address().getAddress(street,city));
    }
}
```

1. I only
2. II only
3. I and II
4. None of the above

36. If a try statement has catch blocks for both `ClassCastException` and `RuntimeException`, then which of the following statements is correct?

1. The catch block for `ClassCastException` must appear before the catch block for `RuntimeException`.
2. The catch block for `RuntimeException` must appear before the catch block for `ClassCastException`.
3. The catch blocks for these two exception types can be declared in any order.
4. A try statement cannot be declared with these two catch block types because they are incompatible.

37. Which of the following is the best scenario to use an exception?

1. The computer caught fire.
2. The code does not compile.
3. A caller passes invalid data to a method.
4. A method finishes sooner than expected.

38. What is the output of the following application?

```
package body;
class Organ {
    public void operate() throws RuntimeException {
        throw new RuntimeException("Not supported");
    }
}
public class Heart extends Organ {
    public void operate() throws Exception {
        System.out.print("beat");
    }
    public static void main(String... cholesterol) throws Exception {
        try {
            new Heart().operate();
        } finally {
        }
    }
}
```

1. beat
2. Not supported

3. The code does not compile.
4. The code compiles but a stack trace is printed at runtime.

39. Which statement about the following exception statement is correct?

```
throw new NullPointerException();
```

1. The code where this is called must include a try-catch block that handles this exception.
2. The method where this is called must declare a compatible exception.
3. This exception cannot be handled.
4. This exception can be handled with a try-catch block or ignored altogether by the surrounding method.

40. What is the output of the following application?

```
package clothing;
public class Coat {
    public Long zipper() throws Exception {
        try {
            String checkZipper = (String)new Object();
        } catch (Exception e) {
            throw RuntimeException("Broken!");
        }
        return null;
    }
    public static void main(String... warmth) {
        try {
            new Coat().zipper();
            System.out.print("Finished!");
        } catch (Throwable t) {}
    }
}
```

1. Finished!
2. Finished!, followed by a stack trace
3. The application does not produce any output at runtime.
4. The code does not compile.

41. Given the following application, which type of exception will be printed in the stack trace at runtime?

```
package carnival;
public class WhackAnException {
    public static void main(String... hammer) {
        try {
            throw new ClassCastException();
        } catch (IllegalArgumentException e) {
            throw new IllegalArgumentException();
        } catch (RuntimeException e) {
            throw new NullPointerException();
        } finally {
            throw new RuntimeException();
        }
    }
}
```

1. IllegalArgumentException
2. NullPointerException
3. RuntimeException
4. The code does not compile.

42. Which of these method signatures is allowed in a class implementing the Outfielder interface?

```
class OutOfBoundsException extends BadCatchException {}
class BadCatchException extends Exception {}

public interface Outfielder {
    public void catchBall() throws OutOfBoundsException;
}
```

1. public int catchBall() throws OutOfBoundsException
2. public int catchBall() throws BadCatchException
3. public int catchBall() throws Exception
4. None of the above

43. What is the output of the following application?

```
package city;
public class Street {
    public static void dancing() throws RuntimeException {
        try {
            throw new IllegalArgumentException();
        } catch (Error) {
            System.out.print("Unable!");
        }
    }
    public static void main(String... count) throws RuntimeException
        dancing();
    }
}
```

1. Unable!
2. The application does not produce any output.
3. The application compiles but produces a stack trace at runtime.
4. The code does not compile.

44. What is the result of compiling and running the following application?

```
package castles;
class DragonException extends Exception {}
public class Lair {
    public void openDrawbridge() throws Exception { // r1
        try {
            throw new Exception("This Exception");
        } catch (RuntimeException e) {
            throw new DragonException(); // r2
        } finally {
            throw new RuntimeException("Or maybe this one");
        }
    }
    public static void main(String[] moat) throws Exception {
        new Lair().openDrawbridge(); // r3
    }
}
```

1. The code does not compile because of line r1.
2. The code does not compile because of line r2.
3. The code does not compile because of line r3.
4. The code compiles, but a stack trace is printed at runtime.

45. If a try statement has catch blocks for both `IllegalArgumentException` and `ClassCastException`, then which of the following statements is correct?

1. The catch block for `IllegalArgumentException` must appear before the catch block for `ClassCastException`.
2. The catch block for `ClassCastException` must appear before the catch block for `IllegalArgumentException`.
3. The catch blocks for these two exception types can be declared in any order.
4. A try statement cannot be declared with these two catch block types because they are incompatible.

46. What is the output of the following application?

```
package broken;
class Problem implements RuntimeException {}
public class BiggerProblem extends Problem {
    public static void main(String uhOh[]) {
        try {
            throw new BiggerProblem();
        } catch (BiggerProblem re) {
            System.out.print("Problem");
        } catch (Problem e) {
            System.out.print("Handled");
        } finally {
            System.out.print("Fixed!");
        }
    }
}
```

1. Problem?Fixed!
2. Handled.Fixed!
3. Problem?Handled.Fixed!

4. The code does not compile.

47. What is the output of the following application?

```
package lighting;
interface Source {
    void flipSwitch() throws Exception;
}
public class LightBulb implements Source {
    public void flipSwitch() {
        try {
            throws new RuntimeException("Circuit Break!");
        } finally {
            System.out.print("Flipped!");
        }
    }
    public static void main(String... electricity) throws Throwable {
        final Source bulb = new LightBulb();
        bulb.flipSwitch();
    }
}
```

1. A stack trace for a RuntimeException
2. Flipped!, followed by a stack trace for a RuntimeException
3. The code does not compile because flipSwitch() is an invalid method override.
4. The code does not compile for another reason.

48. Given an application that hosts a website, which of the following would most likely result in a java.lang.Error being thrown?

1. Two users try to register an account at the same time.
2. The application temporarily loses connection to the network.
3. A user enters their password incorrectly.
4. The application runs out of memory.

49. Given that FileNotFoundException is a subclass of IOException, what is the output of the following application?

```
package storage;
import java.io.*;
public class Backup {
    public void performBackup() {
        try {
            throw new IOException("Disk not found");
        } catch (Exception e) {
            try {
                throw new FileNotFoundException("File not found");
            } catch (FileNotFoundException e) { // z1
                System.out.print("Failed");
            }
        }
    }
    public static void main(String... files) {
        new Backup().performBackup(); // z2
    }
}
```

1. Failed
2. The application compiles but a stack trace is printed at runtime.
3. The code does not compile because of line z1.
4. The code does not compile because of line z2.

50. What is the output of the following application?

```
package bed;
public class Sleep {
    public static void snore() {
        try {
            String sheep[] = new String[3];
            System.out.print(sheep[3]);
        } catch (RuntimeException e) {
            System.out.print("Awake!");
        } finally {
            throw new Exception(); // x1
        }
    }
    public static void main(String... sheep) { // x2
        new Sleep().snore(); // x3
    }
}
```

1. Awake!, followed by a stack trace
2. The code does not compile because of line x1.
3. The code does not compile because of line x2.
4. The code does not compile because of line x3.



PREV

[Chapter 7 Working with Inheritance](#)

NEXT

[Chapter 9 Working with Selected Classes from the Java API](#)