



OCA / OCP Java SE 8 Programmer Practice Tests

PREV
Chapter 15 Java Stream API

NEXT



Chapter 17 Use Java SE 8 Date/Time API

Chapter 16 Exceptions and Assertions

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

- ✓ **Exceptions and Assertions**
- Use try-catch and throw statements
- Use catch, multi-catch, and finally clauses
- Use Autoclose resources with a try-with-resources statement
- Create custom exceptions and Auto-closeable resources
- Test invariants by using assertions

1. If a try, a catch, and a finally statement are used together but no exception is generated, which blocks are executed and in which order?

1. try
2. try, catch
3. try, catch, finally
4. try, finally

2. Fill in the blanks: A try statement _____ a catch or a finally block, while a try-with-resources statement _____.

1. is not required to contain, is not required to contain either
2. is not required to contain, must contain one of them
3. must contain, is not required to contain either
4. must contain, must contain a catch block

3. What is the output of the following application?

```

package park;

class LostBallException extends Exception {}
public class Ball {
    public void toss() throw LostBallException {
        throw new ArrayStoreException();
    }
    public static void main(String[] bouncy) {
        try {
            new Ball().toss();
        } catch (Throwable e) {
            System.out.print("Caught!");
        }
    }
}

```

1. Caught!

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

3. The code does not compile because `ArrayStoreException` is not handled or declared in the `toss()` method.
4. The code does not compile for a different reason.

4. Which symbol(s) can be used to separate exception types in a multi-catch statement?

1. &

2. |

3. ||

1. II only

2. III only

3. II and III

4. I, II, and III

5. What is the result of executing the following application with assertions enabled?

```

1: package ice;
2: public class Igloo {
3:     public static void main(String[] bricks) {
4:         int flakes = 10;
5:         double assert = 7.0;
6:         assert (true : "");
7:         assert flakes++>5;
8:     }
9: }
```

1. It throws an `AssertionError` at runtime.
2. It prints nothing at runtime.
3. Exactly one line of code does not compile.
4. Two lines of code do not compile.

6. Which of the following classes is a checked exception?

1. `java.lang.Error`

2. `java.lang.IllegalStateException`

3. `java.text.ParseException`

4. `java.lang.RuntimeException`

7. How many constructors in `WhaleSharkException` compile in the following class?

```

package friendly;
public class WhaleSharkException extends Exception {
    public WhaleSharkException() {
        super("Friendly shark!");
    }

    public WhaleSharkException(String message) {
        super(new Exception(new WhaleSharkException()));
    }

    public WhaleSharkException(Exception cause) {}
}
```

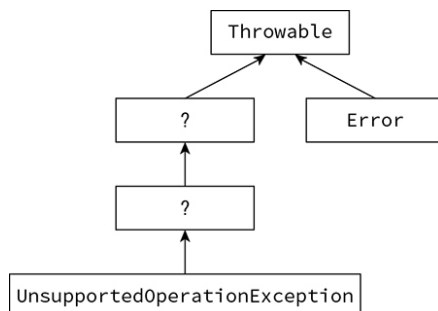
1. None

2. One

3. Two

4. Three

8. Given the following class diagram, which two classes are missing in the hierarchy starting with the bottom and going upward?



1. IOException, Exception
2. RuntimeException, Exception
3. IllegalArgumentException, RuntimeException
4. IllegalStateException, RuntimeException

9. How many lines of text does the following program print?

```

package lighting;
import java.io.IOException;
public class Light {
    public void turnOn() throws IOException {
        new IOException("Not ready");
    }
    public static void main(String[] b) throws Exception {
        try {
            new Light().turnOn();
        } catch (RuntimeException b) { // y1
            System.out.println(b);
            throw new IOException(); // y2
        } finally {
            System.out.println("complete");
        }
    }
}
  
```

1. One
2. Two
3. The code does not compile because of line y1.
4. The code does not compile because of line y2.

10. Which statement, when inserted into the main() method of a program, guarantees an AssertionError will be thrown at runtime?

1. assert(0,"Invalid");
2. assert 0==1;
3. assert 0==0;
4. None of the above

11. What is the output of the following application?

```

package paper;
import java.io.Closeable;
public class PrintCompany {
    class Printer implements Closeable { // r1
        public void print() {
            System.out.println("This just in!");
        }
        public void close() {}
    }
    public void printHeadlines() {
        try {Printer p = new Printer()} { // r2
            p.print();
        }
    }
    public static void main(String[] headlines) {
        new PrintCompany().printHeadlines(); // r3
    }
}
  
```

1. This just in!
2. The code does not compile because of line r1.
3. The code does not compile because of line r2.
4. The code does not compile because of line r3.

12. Which statement about try-with-resources is not true?

1. If the try block and close() method both throw an exception, the one thrown by the close() method is suppressed.
2. A catch block is not required.
3. If more than one resource is used, the resources are closed in the order they were created.
4. Parentheses are used for the resource declaration section, even if more than one resource is used.

13. How many lines of text does the following program print?

```
package bee;
class SpellingException extends RuntimeException {}
public class SpellChecker {
    public final static void main(String... participants) {
        try {
            if(!"cat".equals("kat")) {
                new SpellingException();
            }
        } catch (SpellingException | NullPointerException e) {
            System.out.println("Spelling problem!");
        } catch (Exception e) {
            System.out.println("Unknown Problem!");
        } finally {
            System.out.println("Done!");
        }
    }
}
```

1. One
2. Two
3. Three
4. The code does not compile.

14. Which exception classes, when inserted into the blank in the Problems class, allow the code to compile?

```
package more;
class MissingMoneyException extends Exception {}
class MissingFoodException extends Exception {}
public class Problems {
    public void doIHaveAProblem() throws MissingMoneyException,
        System.out.println("No problems");
    }
    public static void main(String[] lots) throws _____{
        try {
            final Problems p = new Problems();
            p.doIHaveAProblem();
        } catch (Exception e) {
            throw e;
        }
    }
}
```

1. Exception
2. MissingMoneyException
3. MissingMoneyException, MissingFoodException

1. I only
2. III only
3. I and III
4. I, II, and II

15. Which statement about Closeable and AutoCloseable is true?

1. AutoCloseable extends Closeable.
2. The close() method in a class that implements AutoCloseable cannot throw an IOException.
3. The close() method in a class that implements Closeable cannot throw an Exception.
4. There is no difference; one was added for backward compatibility.

16. Which expression, when inserted into the blank in the following class, allows the code to compile?

```

package sun;
import java.io.*;
public class Beach {
    class TideException extends Exception {}
    public void surf() throws RuntimeException {
        try {
            throw new TideException();
        } catch (_____) {}
    }
}

```

1. Exception a | RuntimeException f
2. IllegalStateException | TideException t
3. TideException | IOException i
4. TideException | Exception x

17. Which statement about a multi-catch statement is true?

1. The exception types must be ordered from broadest to narrowest.
2. The exception types must be ordered from narrowest to broadest.
3. The variable of a multi-catch block with more than one exception cannot be reassigned within the block.
4. The variable of a multi-catch block with one exception type cannot be reassigned within the block.

18. Given the following class, how many lines contain compilation errors?

```

package move;
interface Closing {
    void close() throws Exception;
}
class Shelf implements Closing {
    public void close() throws Exception {}
}
public class Step {
    static {
        try {Shelf shelf = new Shelf(); {
            throws new IllegalArgumentException();
        } catch (Exception e) {
        } catch (IllegalArgumentException e) {
        } finally {
            shelf.close();
        }
    }
}

```

1. None
2. Two
3. Three
4. Four

19. Which of the following is not true of using a try-with-resources statement?

1. Associated catch blocks are run before the declared resources have been closed.
2. It is compatible with all classes that implement the AutoCloseable interface.
3. It is compatible with all classes that implement the Closeable interface.
4. It shortens the amount of code a developer must write.

20. Assuming the following application is executed with assertions enabled, what is the result?

```

package input;
public class DataIntegrity {
    private int score;
    public DataIntegrity() {
        super();
        DataIntegrity.this.score = 5;
    }
    public static void main(String[] books) {
        final DataIntegrity johnny5 = new DataIntegrity();
        assert(johnny5.score>2) : johnny5.score++;
        assert johnny5.score>=5 : System.out.print("No input");
        System.out.print("Made it!");
    }
}

```

1. An AssertionError is thrown with a message of 5.
2. An AssertionError is thrown with a message of No input.
3. Made it! is printed.
4. The code does not compile.

21. Which of the following classes is an unchecked exception?

1. java.io.IOException
2. java.io.NotSerializableException
3. java.sql.SQLException
4. java.util.MissingResourceException

22. What is the result of compiling and executing the following class?

```
package wind;
public class Storm {
    public static void main(String... rain) throws Exception {
        try (final AutoCloseable weatherTracker = new AutoCloseable() {
            public void close() throws RuntimeException {}
        }) {
            System.out.println(weatherTracker.toString());
        } catch (Exception e) {
            if(weatherTracker != null) {
                weatherTracker.close();
            }
        } finally {
            System.out.println("Storm gone");
        }
    }
}
```

1. It prints one line.
2. It prints two lines.
3. It does not compile due to an error in the declaration of the weatherTracker resource.
4. It does not compile for a different reason.

23. Which of the following is not a command that enables or disables assertions at runtime?

1. -di
2. -disableassertions
3. -ea
4. -enableassertions

24. What is the output of the following application?

```
package signlanguage;
import java.io.Closeable;
class ReadSign implements Closeable {
    public void close() {}
    public String get() {return "Hello";}
}
class MakeSign implements AutoCloseable {
    public void close() {}
    public void send(String message) {
        System.out.print(message);
    }
}
public class Translate {
    public static void main(String... hands) {
        try (ReadSign r = new ReadSign();
            MakeSign w = new MakeSign();) {
            w.send(r.get());
        }
    }
}
```

1. Hello
2. The code does not compile because of the ReadSign class.
3. The code does not compile because of the try-with-resources statement.
4. None of the above

25. What is the output of the following application?

```

package what;
class FunEvent implements AutoCloseable {
    public void close() {
        System.out.print("1");
    }
}
public class Happening {
    public static void main(String... lots) {
        try (FunEvent f = new FunEvent()) {
            System.out.print("2");
            throw new ArithmeticException();
        } catch (Exception e) {
            System.out.print("3");
        } finally {
            System.out.print("4");
        }
    }
}

```

1. 214
2. 2134
3. 2314
4. The code does not compile.

26. Which statement best describes how a class that implements the `AutoCloseable` interface should be written?

1. The `close()` method is optional since the `AutoCloseable` interface defines a default implementation.
2. The `close()` method should avoid modifying data after it has been run once.
3. The `close()` method should not throw any exceptions.
4. The `close()` method should return a status code.

27. Which statement about the following program is correct?

```

package dogpark;
public class Fetch {
    public int play(String dogName) throws Exception {
        try {
            throw new RuntimeException(dogName);
        } catch (Exception e) {
            throw new RuntimeException(e);
        }
    }
    public static final void main(String[] ball) throws RuntimeExcep
        new Fetch().play("Webby");
        new Fetch().play("Georgette");
    }
}

```

1. The program prints one exception at runtime.
2. The program prints two exceptions at runtime.
3. The class does not compile because of the `play()` method.
4. The class does not compile because of the `main()` method.

28. Which of the following is not a good use of assertions?

1. Check method post conditions.
2. Modify local variables.
3. Test control flow invariants.
4. Validate class invariants.

29. Which statement about the following application is correct?

```

package highway;
import java.io.*;
class CarCrash extends RuntimeException {}
public class Car {
    public static void main(String[] seatbelts) throws Exception {
        try {
            throw new IOException("Auto-pilot error");
        } catch (Exception | CarCrash e) { // w2
            throw e;
        } catch (Exception a) { // w3
            throw a;
        }
    }
}

```

-
1. The code does not compile because of line w1.
 2. The code does not compile because of line w2.
 3. The code does not compile because of line w3.
 4. The code compiles and runs without issue.

30. Which statements about the following classes are true?

```
public class Dopey extends Grumpy {}
public class Grumpy extends Exception {}
public class Happy extends IOException {}
public class Sleepy extends IllegalStateException {}
public class Sneezzy extends Throwable {}
```

1. Four of the classes are checked exceptions.
 2. Two of the classes are unchecked exceptions.
 3. None of the class declarations contain any compilation errors.
1. I only
 2. I and III
 3. II and III
 4. I, II, and III

31. What is the output of the following application?

```
package vortex;
class TimeException extends Exception {}
class TimeMachine implements AutoCloseable {
    int v;
    public TimeMachine(int v) {this.v = v;}
    public void close() throws Exception {
        System.out.print(v);
    }
}
public class TimeTraveler {
    public static void main(String[] twelve) {
        try (TimeMachine timeSled = new TimeMachine(1);
            TimeMachine delorean = new TimeMachine(2);
            TimeMachine tardis = new TimeMachine(3)) {
            } catch (TimeException e) {
                System.out.print(4);
            } finally {
                System.out.print(5);
            }
        }
    }
}
```

1. 1235
2. 3215
3. 41235
4. The code does not compile.

32. Which expression, when inserted into the blank in the following class, allows the code to compile?

```
package music;
public class Bells {
    class Player implements AutoCloseable {
        @Override public void close() throws RingException {}
    }
    class RingException extends Exception {
        public RingException(String message) {}
    }
    public static void main(String[] notes) throws Throwable {
        try (Player p = null) {
            throw new Exception();
        } catch (Exception e) {
        } catch (_____) {
        }
    }
}
```

1. Error r
2. IllegalStateException b
3. RingException p
4. The code does not compile regardless of the expression used.

33. Given the following two variables, which assertion statement compiles successfully?

```
int age = 22;
final String name = "Josephine";
```

1. `assert (age=2);`
2. `assert age!=age : (1<age ? "Error" : 10);`
3. `assert name.equals("") : () -> "Oops";`
4. `assert name.length()<(long)age : return "Mistake";`

34. Which statement about the following program is true?

```
package tag;
class MissedCallException extends Exception {}
public class Phone {
    static void makeCall() throws RuntimeException {
        throw new ArrayIndexOutOfBoundsException("Call");
    }
    public static void main(String[] messages) {
        try {
            makeCall();
        } catch (MissedCallException e) {
            throw new RuntimeException("Voicemail");
        } finally {
            throw new RuntimeException("Text");
        }
    }
}
```

1. An exception is printed at runtime with `Call` in the message.
2. An exception is printed at runtime with `Voicemail` in the message.
3. An exception is printed at runtime with `Text` in the message.
4. The code does not compile.

35. Which statement about the following program is correct?

```
package fairy;
public class Tale {
    class BearException extends RuntimeException {}
    class WolfException extends RuntimeException {}
    class DragonException extends RuntimeException {}
    public int tellStory() {
        try {} catch (BearException d) {
            d = new RuntimeException();
            throw d;
        } catch (WolfException | DragonException e) {
            e = new RuntimeException();
            throw e;
        }
        return 3;
    }
    public static void main(String... wand) throws RuntimeException {
        new Tale().tellStory();
    }
}
```

1. The class compiles and does not print anything at runtime.
2. The code does not compile solely due to the first catch block in `tellStory()`.
3. The code does not compile solely due to the second catch block in `tellStory()`.
4. The code does not compile due to errors in both catch blocks in `tellStory()`.

36. What is the output of the following application?

```
package classical;
import java.io.*;
class OutOfTuneException extends Exception {
    OutOfTuneException(String message) { super(message); }
}
public class Piano {
    public void play() throws OutOfTuneException, FileNotFoundException {
        throw new OutOfTuneException("Sour note!");
    }
    public static void main(String... keys) throws OutOfTuneException {
        final Piano piano = new Piano();
        try {
            piano.play();
        } catch (Exception e) {
            throw e;
        }
    }
}
```

```

        } finally {
            System.out.println("Song finished!");
        }
    }
}

```

1. Song finished!
2. An exception is printed with Source note! in the stack trace.
3. Both of the above
4. None of the above

37. Given the following class, which command causes the class to throw an `AssertionError` at runtime?

```

public class Falcon extends Exception {
    private int parsec = 12;
    public Falcon(String name) {
        super(name);
    }
    public static void main(String[] aluminum) {
        assert new Falcon(null).parsec < 12;
    }
}

```

1. `java Falcon`
2. `java -ea -da:Falcon Falcon`
3. `java -da -ea:Falcon Falcon`
4. The code does not compile.

38. What is the output of the following application?

```

package db;
import java.io.*;
import java.sql.*;
public class DatabaseHelper {
    static class MyDatabase implements Closeable {
        public void close() throws SQLException {
            System.out.print("2");
        }
        public void write(String data) {}
        public String read() {return null;}
    }
    public static void main(String... files) throws Exception {
        try (MyDatabase myDb = new MyDatabase()) {
            // TODO: Decide what to read/write
        } finally {
            System.out.print("1");
        }
    }
}

```

1. 12
2. 21
3. The code does not compile because of the `MyDatabase` class.
4. The code does not compile because of the try-with-resources statement.

39. How many lines of text does the following program print?

```

package tron;
class DiskPlayer implements AutoCloseable {
    public void close() throws Exception {}
}
public class LightCycle {
    public static void main(String... bits) {
        try (DiskPlayer john = new DiskPlayer()) {
            System.out.println("ping");
        } finally {
            System.out.println("pong");
        }
        System.out.println("return");
    }
}

```

1. One
2. Two
3. Three
4. The code does not compile.

40. Given the application below, what is the name of the class printed at line e1?

```
package canyon;
final class FallenException extends Exception {}
final class HikingGear implements AutoCloseable {
    @Override public void close() throws Exception {
        throw new FallenException();
    }
}
public class Cliff {
    public final void climb() throws Exception {
        try (HikingGear gear = new HikingGear()) {
            throw new RuntimeException();
        }
    }
    public static void main(String... rocks) {
        try {
            new Cliff().climb();
        } catch (Throwable t) {
            System.out.println(t); // e1
        }
    }
}
```

1. canyon.FallenException
2. java.lang.RuntimeException
3. The code does not compile.
4. The code compiles, but the answer cannot be determined until runtime.



◀ PREV
Chapter 15 Java Stream API

NEXT ▶
Chapter 17 Use Java SE 8 Date/Time API