



OCA / OCP Java SE 8 Programmer Practice Tests



PREV

Chapter 17 Use Java SE 8 Date/Time API



NEXT



Chapter 19 Java File I/O (NIO.2)

Chapter 18

Java I/O Fundamentals

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

- **✓ Java I/O Fundamentals**
 - Read and write data from the console
 - Use `BufferedReader`, `BufferedWriter`, `File`, `FileReader`, `FileWriter`, `FileInputStream`, `FileOutputStream`, `ObjectOutputStream`, `ObjectInputStream`, and `PrintWriter` in the `java.io` package.
1. Fill in the blanks: `Writer` is _____ that related stream classes _____.
 1. a concrete class, extend
 2. an abstract class, extend
 3. an interface, extend
 4. an interface, implement
 2. Which of the following methods is defined in `java.io.File`?
 1. `createDirectory()`
 2. `getLength()`
 3. `listFile()`
 4. `renameTo()`
 3. Which method in `InputStream` can be used in place of calling `skip(1)`?
 1. `jump()`
 2. `mark()`
 3. `read()`
 4. `reset()`
 4. Which methods are classes that implement `java.io.Serializable` required to implement?
 1. `deserialize()`
 2. `serial()`
 3. `serialize()`
 4. None of the above
 5. Fill in the blanks: Given a valid `Console` instance, `reader()` returns a _____, while `writer()` returns a _____.

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

2. PrintReader, Writer
3. Reader, Writer
4. StringReader, Writer

6. Assuming the file path referenced in the following class is accessible and able to be written, what is the output of the following program?

```
package alarm;
import java.io.*;
public class Smoke {
    public void sendAlert(File fn) {
        try(BufferedWriter w = new BufferedWriter(new FileOutputStream(
            w.write("ALERT!"));
            w.flush();
            w.write('!');
            System.out.print("1");
        } catch (IOException e) {
            System.out.print("2");
        } finally {
            System.out.print("3");
        }
    }
    public static void main(String[] testSignal) {
        new Smoke().sendAlert(new File("alarm.txt"));
    }
}
```

1. 3
2. 13
3. 23
4. The code does not compile.

7. Which class is used to read information about a directory within the file system?

1. java.io.File
2. java.io.Directories
3. java.io.Directory
4. java.io.Path

8. Which of the following is a high-level stream class that can only be used to wrap a low-level stream?

1. FileOutputStream
2. FileReader
3. ObjectInputStream
4. PrintWriter

9. Assume the file prime6.txt exists and contains the first six prime numbers as bytes: 2, 3, 5, 7, 11, 13. What is the output of the following application?

```
package numbers;
import java.io.*;
public class PrimeReader {
    public static void main(String[] real) throws Exception {
        try (InputStream is = new FileInputStream("prime6.txt")) {
            is.skip(1);
            is.read();
            is.skip(1);
            is.read();
            is.mark(4);
            is.skip(1);
            is.reset();
            System.out.print(is.read());
        }
    }
}
```

1. 11
2. 13
3. The code does not compile.
4. The code compiles but throws an exception at runtime.

10. Fill in the blanks: For a given file, the absolute is the path from the _____ to the file, while the relative path is the path from the _____ to the file.

1. current directory, current working directory
2. parent directory, temporary directory
3. root directory, current working directory
4. root directory, parent directory

11. Which statement best describes the following two methods?

```
public void writeSecret1() throws IOException {
    final Writer w = new BufferedWriter(
        new FileWriter("dont.open"));
    w.write("Secret passcode");
    w.close();
}
public void writeSecret2() throws IOException {
    try(final Writer w = new BufferedWriter(
        new FileWriter("dont.open"))) {
        w.write("Secret passcode");
    }
}
```

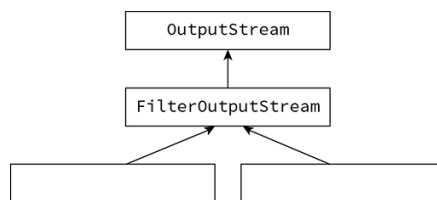
1. Both methods compile and are equivalent to each other.
2. Neither method compiles.
3. Only one of the methods compiles.
4. The methods compile, but one method may lead to a resource leak.

12. What is the result of compiling and executing the following program?

```
package vacation;
import java.io.*;
import java.util.*;
public class Itinerary {
    private List<String> activities = new ArrayList<>();
    private static Itinerary getItinerary(String name) {
        return null;
    }
    public static void printItinerary() throws Exception {
        Console c = new Console();
        final String name = c.readLine("What is your name?");
        final Itinerary stuff = getItinerary(name);
        stuff.activities.forEach(s -> c.printf(s));
    }
    public static void main(String[] holidays) throws Exception {
        printItinerary();
    }
}
```

1. The code does not compile.
2. The code compiles and prints a NullPointerException at runtime.
3. The code compiles but does not print anything at runtime.
4. None of the above

13. Given the following diagram, which two classes can be placed in the blank boxes?



1. BufferedOutputStream and PrintStream
2. BufferedOutputStream and PrintOutputStream
3. ByteArrayOutputStream and Stream
4. FileOutputStream and OutputStream

14. Let's say we want to write an instance of Cereal to disk, having a name value of CornLoops. What is the value of name after this object has been read using the ObjectOutputStream's readObject() method?

```

package breakfast;
public class Cereal {
    private String name = "CocoaCookies";
    private transient int sugar;
    public Cereal() {
        super();
        this.name = "CaptainPebbles";
    }
    {
        name = "SugarPops";
    }
    public String getName() { return name; }
    public void setName(String name) {
        this.name = name;
    }
    public int getSugar() { return sugar; }
    public void setSugar(int sugar) {
        this.sugar = sugar;
    }
}

```

1. CaptainPebbles
 2. CornLoops
 3. SugarPops
 4. None of the above
15. Which statement best describes the difference between a `Writer` and an `OutputStream` class?
1. Only one of them can write text or character data.
 2. Only one of them has built-in methods for writing character data.
 3. Only one of them has a `flush()` method to force the data to be written out.
 4. One uses a byte array to process character data more efficiently.
16. What is the output of the following application? It is safe to assume the directories referenced in the class do not exist prior to the execution of the program and that the file system is available and able to be written.

```

package job;
import java.io.*;
public class Resume {
    public void resetWorkingDirectory() throws Exception {
        File f1 = new File("/templates/proofs");
        f1.mkdirs();
        File f2 = new File("/templates");
        f2.mkdir(); // k1
        new File(f2,"draft.doc").createNewFile();
        f1.delete();
        f2.delete(); // k2
    }
    public static void main(String... leads) {
        try {
            new Resume().resetWorkingDirectory();
        } catch (Exception e) {
            new RuntimeException(e);
        }
    }
}

```

1. Line k1 does not compile or triggers an exception at runtime.
 2. Line k2 does not compile or triggers an exception at runtime.
 3. The code compiles and runs without printing an exception.
 4. None of the above
17. Given the following class, three of the values ensure it runs properly on various different systems. Which value does not?

```

package magic;
import java.io.*;
public class Store {
    private final String directory;
    public Store(String directory) {
        this.directory = directory;
    }
    public File getDatabaseFolder(String file) {
        return new File(directory + _____ + file);
    }
}

```

1. `java.io.File.separator`
2. `new File(new String()).separatorChar`

3. `System.getProperty("file.separator")`
4. `System.getProperty("path.separator")`

18. How many compilation errors does the following class contain?

```
package hero;
import java.io.*;
public class Guitar {
    public void readMusic(File f) {
        try (BufferedReader r = new BufferedReader(FileReader(f))) {
            final String music = null;
            try {
                while((music = r.readLine()) != null)
                    System.out.println(music);
            } catch (IOException e) {}
        } catch (FileNotFoundException e) {
            throw new RuntimeException(e);
        } finally {}
    }
}
```

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

19. What is the difference between the two Console methods, `format()` and `printf()`?

1. One of them takes an optional list of arguments; the other does not.
2. One of them takes `String` as input; the other takes an `Object`.
3. There is no difference between the two methods.
4. Trick question! `printf()` is not defined in `Console`.

20. Let's say you want to write a lot of text data to a file in an efficient manner. Which two `java.io` stream classes are best to use?

1. `FileOutputStream` and `BufferedOutputStream`
2. `FileOutputStream` and `FileBufferedWriter`
3. `FileWriter` and `BufferedWriter`
4. `ObjectOutputStream` and `BufferedWriter`

21. Assume the file referenced in the `StudentManager` class exists and contains data. Which statement about the following class is correct?

```
package school;
import java.io.*;
class Student implements Serializable {}
public class StudentManager {
    public static void main(String[] grades) {
        try (ObjectInputStream ios = new ObjectInputStream(
            new FileInputStream(new File("C://students.data")))) {
            Student record;
            while((record = (Student)ios.readObject()) != null) {
                System.out.print(record);
            }
        } catch (EOFException e) {}
        } catch (Exception e) {
            throw new RuntimeException(e);
        }
    }
}
```

1. The code does not compile.
2. The code compiles but prints an exception at runtime.
3. The program runs and prints all students in the file.
4. The program runs but may only print some students in the files.

22. Which `java.io` class does not have a complementary input stream?

1. `BufferedOutputStream`
2. `BufferedWriter`
3. `FileWriter`

4. PrintWriter

23. Assuming the path `/Earth` does not exist within the file system, what is the output of the following program?

```
package center;
import java.io.*;
public class Journey {
    public static void main(String[] dig) {
        File file = new File("/Earth");
        System.out.print(file.getParent()
            + " - "
            + file.getParent().getParent()); } }
```

1. `/ - /`
2. `/ - null`
3. The code does not compile.
4. The code compiles but throws an exception at runtime.

24. Which statements about executing the following program are true?

```
package test;
import java.io.*;
public class Turing {
    public static void main(String... robots) {
        Console c = System.console();
        final String response = c.readLine("Are you human?");
        System.err.print(response);
    }
}
```

1. The program may ask the user a question and print the response to the error stream.
 2. The program may throw a `NullPointerException` at runtime.
 3. The program may wait indefinitely.
1. I
 2. I and III
 3. II and III
 4. I, II, and III

25. Which of the following statements about the `deleteTree()` method is correct?

```
public void deleteTree(File f) {
    if(!f.isDirectory())
        f.delete();
    else {
        Stream.of(f.listFiles())
            .forEach(s -> deleteTree(s));
        f.deleteDirectory();
    }
}
```

1. It compiles and is capable of deleting a directory tree.
2. If one line were modified, it would be capable of deleting a directory tree.
3. If two lines were modified, it would be capable of deleting a directory tree.
4. None of the above

26. Which of the following is not a built-in stream in Java?

1. `System.err`
2. `System.in`
3. `System.info`
4. `System.out`

27. Assuming the file path referenced in the following class is accessible and able to be written, what is the output of the following program?

```
package store;
import java.io.*;
public class Furniture {
    public final static void main(String... inventory) throws Except:
```

```

        Writer w = new FileWriter("couch.txt");
        try (BufferedWriter bw = new BufferedWriter(w)) {
            bw.write("Blue coach on Sale!");
        } finally {
            w.flush();
            w.close();
        }
        System.out.print("Done!");
    }
}

```

1. Done!
2. The code does not compile for one reason.
3. The code does not compile for two reasons.
4. The code compiles but throws an exception at runtime.

28. Given an instance of `Console c`, which of the following method calls is not a way to read input from the user?

1. `c.reader().read()`
2. `c.reader().readLine()`
3. `c.readLine()`
4. `c.readPassword()`

29. The `copyPidgin()` method is used to copy the contents of one file to another. Which statement about the implementation is correct?

```

package birds;
import java.io.*;
public class Pidgin {
    public void copyPidgin(File s, File t) throws Exception {
        try(InputStream is = new FileInputStream(s);
            OutputStream os = new FileOutputStream(t)) {
            byte[] data = new byte[123];
            int chirps;
            while((chirps = is.read(data))>0) {
                os.write(data);
            }
        }
    }
}

```

1. The class does not compile because `read(byte[])` and `write(byte[])` can only be called on `BufferedInputStream` and `BufferOutputStream`, respectively.

2. The method correctly copies the contents of all files.
3. The method correctly copies the contents of some files.
4. The method will always throw an exception at runtime because the data array size is not a power of 2.

30. Using what you know about `java.io` stream class names, what would a nonexistent class named `BufferedReaderFileReader` most likely be used for?

1. Reading a small text file from a remote network
2. Reading an image from disk
3. Reading large text files from a file system
4. Reading serialized data from disk

31. What is the output of the following application?

```

package factory;
import java.io.*;
public class WidgetProcessor {
    public int getWidgetNumber(byte[] data) throws Exception {
        try (InputStream is = new ByteArrayInputStream(data)) {
            is.read(new byte[2]);
            if(!is.markSupported()) return -1;
            is.mark(5);
            is.read();is.read();
            is.skip(3);
            is.reset();
            return is.read();
        }
    }
    public static void main(String... sprockets) throws Exception {
        final WidgetProcessor p = new WidgetProcessor();
        System.out.print(p.getWidgetNumber(new byte[] {1,2,3,4,5,6,7})
    }
}

```

-
1. 3
 2. 5
 3. 7
 4. An exception is thrown at runtime.

32. Assuming the working directory is accessible, empty, and able to be written, how many file system objects does the following class create?

```

1: package kitchen;
2: import java.io.*;
3: public class Bakers {
4:     public static void main(String... tooMany) throws IOException
5:     {
6:         File cake = new File("cake.txt");
7:         Writer pie = new FileWriter("pie.txt");
8:         pie.flush();
9:         new File("fudge.txt").mkdirs();
10:    }

```

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

33. Let's say you wanted to read data from a file stored on disk that consists of `String`, `long`, and `Object` values? Given that the file is quite large, you intend to use three classes to achieve this result. Which of the following is not one of the three classes you should use?

1. `BufferedInputStream`
2. `BufferedReader`
3. `FileInputStream`
4. `ObjectInputStream`

34. Which statement best describes the following two methods?

```

public String getNameQuick() throws IOException {
    final BufferedReader r = new BufferedReader(
        new FileReader("saved.name"));
    final String name = r.readLine();
    r.flush();
    return name;
}

public String getNameSafely() throws IOException {
    try {
        final BufferedReader r = new BufferedReader(
            new FileReader("saved.name"));
        final String name = r.readLine();
        r.flush();
        return name;
    }
}

```

1. Both methods compile and are equivalent to each other.
2. Neither method compiles.
3. Only one of the methods compiles.
4. The methods compile, but one method may lead to a resource leak.

35. What is the output of the following application? Assume the `System.console()` is available and the user enters `badxbad` and presses Enter.

```

package hardway;
import java.io.*;
public class InconvenientImplementation {
    public static void main(String... dontDoThis) throws Exception {
        Console c = System.console();
        if(c != null) {
            c.writer().write('P');
            c.writer().write('a');
            c.writer().write('s');
            c.writer().write('s');
            c.writer().flush(); // t1
            int i;
            StringBuilder sb = new StringBuilder();
            while((i = c.reader().read()) != 'x') { // t2
                sb.append((char)i);
            }
        }
    }
}

```

```

        c.writer().format("Result: %s",sb.toString());
    }
}

```

-
1. Result: bad
 2. Line t1 does not compile or triggers an exception at runtime.
 3. Line t2 does not compile or triggers an exception at runtime.
 4. None of the above
36. Why does `Console.readPassword()` return a char array rather than a `String`?
1. It improves performance.
 2. It improves security.
 3. Passwords must be stored as a char array.
 4. `String` cannot hold the individual password characters.
37. Which statement about the following program is true?

```

package mystical;
import java.io.*;
public class Unicorn {
    public void findUnicorns() {
        try(InputStream o = new ObjectInputStream(readBook())) {
            while(o.read() != -1) {
                System.out.println(o.read());
            }
        } catch (Throwable t) {
            throw new RuntimeException(t);
        }
    }
    private InputStream readBook() throws IOException {
        return new BufferedInputStream(new FileReader("magic.book"));
    }
    public static void main(String... horn) {
        new Unicorn().findUnicorns();
    }
}

```

-
1. The code does not compile.
 2. The program prints every byte in the file without throwing an exception.
 3. The program prints every other byte in the file without throwing an exception.
 4. The program throws an `EOFException` when the end of the file is reached.
38. Choose the class that is least likely to be marked `Serializable`.
1. A class that holds data about the amount of rain that has fallen in a given year
 2. A class that manages the memory of running processes in an application
 3. A class that stores information about apples in an orchard
 4. A class that tracks the amount of candy in a gumball machine
39. What is the output of the following application?

```

package cell;
import java.io.*;
public class TextMessage {
    public String receiveText() throws Exception {
        try (Reader r = new FileReader("messages.txt")) {
            StringBuilder s = new StringBuilder();
            int c;
            while((c = r.read()) != -1) {
                s.append((char)c);
                if(r.markSupported()) {
                    r.mark(100);
                    r.skip(10);
                    r.reset();
                }
            }
            return s.toString();
        }
    }
}

```

```

    }
    public void sendText(String message) throws Exception {
        try (Writer w = new FileWriter("messages.txt")) {
            for(int i=0; i<message.length(); i++) {
                w.write(message.charAt(i));
                w.skip(1);
            }
        }
    }
    public static void main(String[] minutes) throws Exception {
        final TextMessage m = new TextMessage();
        m.sendText("You up?");
        System.out.println(m.receiveText());
    }
}

```

1. You up?
2. Y o u u p ?
3. The code does not compile because of the receiveText() method.
4. The code does not compile because of the sendText() method.

40. What is the output of the following program? Assume the file paths referenced in the class exist and are able to be written to and read from.

```

package heart;
import java.io.*;
public class Valve implements Serializable {
    private int chambers = -1;
    private transient Double size = null;
    private static String color;
    public Valve() {
        this.chambers = 3;
        color = "BLUE";
    }

    public static void main(String[] love) throws Throwable {
        try (ObjectOutputStream o = new ObjectOutputStream(
            new FileOutputStream("scan.txt"))) {
            final Valve v = new Valve();
            v.chambers = 2;
            v.size = 10.0;
            v.color = "RED";
            o.writeObject(v);
        }
        new Valve();
        try (ObjectInputStream o = new ObjectInputStream(
            new FileInputStream("scan.txt"))) {
            Valve v = (Valve)o.readObject();
            System.out.print(v.chambers+" "+v.size+" "+v.color);
        }
    }
    { chambers = 4; }
}

```

1. 2,null,RED
2. 2,null,BLUE
3. 3,10.0,RED
4. The code does not compile.

