



OCA / OCP Java SE 8 Programmer Practice Tests



PREV

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

NEXT

[Part II OCP](#)

Chapter 10

OCA Practice Exam

This chapter contains 80 questions and is designed to simulate a real OCA exam. While previous chapters were focused on a specific set of objectives, this chapter covers all of the objectives on the exam. We recommend you take this exam only after you score well on the questions in the individual chapters.

For this chapter, you should try to simulate the real exam experience as much as possible. This means setting aside 150 minutes of uninterrupted time to complete the test, as well as not looking at any reference material while taking the exam. If you don't know an answer to a question, complete it as best you can and move on to the next question, just as you would on a real exam.

Remember, the exam permits writing material, such as a whiteboard. If you do not have a whiteboard handy, you can just use blank sheets of paper and a pencil. If you do well on this test, then you are hopefully ready to take the real exam. With that said, good luck!

1. What is the output if this class is run with `java Indexing cars carts`?

```
public class Indexing {
    public static void main(String... books) {
        StringBuilder sb = new StringBuilder();
        for (String book : books)
            sb.insert(sb.indexOf("c"), book);
        System.out.println(sb);
    }
}
```

1. cars
2. cars carts
3. ccars arts
4. The code does not compile.
5. The code compiles but throws an exception at runtime.

2. Fill in the blanks: The operators `+=`, _____, _____, _____, _____, and `++` are listed in increasing or the same level of operator precedence. (Choose two.)

1. `-`, `+`, `+=`, `--`
2. `%`, `*`, `/`, `+`
3. `=`, `+`, `/`, `*`
4. `^`, `*`, `-`, `==`
5. `*`, `/`, `%`, `--`

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

```
1. public byte getNose(String nose)
2. public void setHead(int head)
3. public String getShoulders()
4. public long isMouth()
5. public void gimmeEars()
6. public boolean isToes()
```

4. Which of the following are true? (Choose two.)

```
20: int[] crossword [] = new int[10][20];
21: for (int i = 0; i < crossword.length; i++)
22:     for (int j = 0; j < crossword.length; j++)
23:         crossword[i][j] = 'x';
24: System.out.println(crossword.size());
```

1. One line needs to be changed for this code to compile.
2. Two lines need to be changed for this code to compile.
3. Three lines need to be changed for this code to compile.
4. If the code is fixed to compile, none of the cells in the 2D array have a value of 0.
5. If the code is fixed to compile, half of the cells in the 2D array have a value of 0.
6. If the code is fixed to compile, all of the cells in the 2D array have a value of 0.

5. Which of the following statements about `java.lang.Error` are most accurate? (Choose two.)

1. An `Error` should be thrown if a file system resource becomes temporarily unavailable.
2. An application should never catch an `Error`.
3. `Error` is a subclass of `Exception`, making it a checked exception.
4. It is possible to catch and handle an `Error` thrown in an application.
5. An `Error` should be thrown if a user enters invalid input.

6. Given a class that uses the following `import` statements, which class would be automatically accessible without using its full package name? (Choose three.)

```
import forest.Bird;
import jungle.tree.*;
import savana.*;
```

1. `forest.Bird`
2. `savana.sand.Wave`
3. `jungle.tree.Huicungo`
4. `java.lang.Object`
5. `forest.Sloth`
6. `forest.apes.bonobo`

7. How many of the following variables represent immutable objects?

```
ArrayList l = new ArrayList();
String s = new String();
StringBuilder sb = new StringBuilder();
LocalDateTime t = LocalDateTime.now();
```

1. None
2. One
3. Two
4. Three
5. Four
6. None of the above—this code doesn't compile.

8. What is the output of the following?

```

Stringbuilder builder = new StringBuilder("Leaves growing");
do {
    builder.delete(0, 5);
} while (builder.length() > 5);
System.out.println(builder);

```

1. Leaves growing
2. ing
3. wing
4. The code does not compile.
5. The code compiles but throws an exception at runtime.

9. What is the output of the following application?

```

package reality;
public class Equivalency {
    public static void main(String[] edges) {
        final String ceiling = "up";
        String floor = new String("up");
        final String wall = new String(floor);
        System.out.print((ceiling==wall)      + " " + (floor==wall))
    }
}

```

1. false false false
2. true true true
3. false true true
4. false false true
5. It does not compile.

10. How many times does the following code print true?

```

1: public class Giggles {
2:     public static void main(String[] args) {
3:         String lol = "lol";
4:         System.out.println(lol.toUpperCase() == lol);
5:         System.out.println(lol.toUpperCase() == lol.toUpperCase(
6:         System.out.println(lol.toUpperCase().equals(lol)));
7:         System.out.println(lol.toUpperCase().equals(lol.toUpperCase(
8:         System.out.println(lol.toUpperCase().equalsIgnoreCase(lo
9:         System.out.println(lol.toUpperCase(
10:             .equalsIgnoreCase(lol.toUpperCase()));
11:     } }

```

1. One
2. Two
3. Three
4. Four
5. Five
6. None. The code does not compile.

11. Which lines can be removed together without stopping the code from compiling and while printing the same output? (Choose three.)

```

14: String race = "";
15: outer:
16: do {
17:     inner:
18:     do {
19:         race += "x";
20:     } while (race.length() <= 4);
21: } while (race.length() < 4);
22: System.out.println(race);

```

1. Lines 15 and 17
2. Lines 15, 16, and 21
3. Line 17
4. Lines 17, 18, and 20
5. Line 20
6. Line 21

12. Which of the following do not compile when filling in the blank?
(Choose two.)

```
long bigNum = _____;
```

1. 1234
2. 1234.0
3. 1234.0L
4. 1234l
5. 1234L
6. 1_234

13. How many lines does this program print?

```
import java.time.*;
public class OnePlusOne {
    public static void main(String... nums) {
        LocalDateTime time = LocalDateTime.of(1, 11);
        while (time.getHour() < 1) {
            time.plusHours(1);
            System.out.println("in loop");
        }
    }
}
```

1. None
2. One
3. Two
4. This is an infinite loop.
5. The code does not compile.

14. What is the result of running the following program?

```
1: package fun;
2: public class Sudoku {
3:     static int[][] game;
4:
5:     public static void main(String args[]) {
6:         game[3][3] = 6;
7:         Object[] obj = game;
8:         obj[3] = 'X';
9:         System.out.println(game[3][3]);
10:    }
11: }
```

1. 6
2. X
3. The code does not compile.
4. The code compiles but throws a NullPointerException at runtime.
5. The code compiles but throws a different exception at runtime.
6. The output is not guaranteed.

15. Which of the following use generics and compile without warnings?
(Choose two.)

1. List<String> a = new ArrayList();
2. List<> b = new ArrayList();
3. List<String> c = new ArrayList<>();
4. List<> d = new ArrayList<>();
5. List<String> e = new ArrayList<String>();
6. List<> f = new ArrayList<String>();

16. Which of the following are true right before the main() method ends?
(Choose two.)

```
public static void main(String[] args) {
    String shoe1 = new String("sandal");
    String shoe2 = new String("flip flop");
    String shoe3 = new String("croc");

    shoe1 = shoe2;
    shoe2 = shoe3;
```

```

        shoe3 = shoe1;
    }

```

1. No objects are eligible for garbage collection.
2. One object is eligible for garbage collection.
3. Two objects are eligible for garbage collection.
4. No objects are guaranteed to be garbage collected.
5. One object is guaranteed to be garbage collected.
6. Two objects are guaranteed to be garbage collected.

17. How many lines of the following application do not compile?

```

package ocean;
class BubbleException extends Exception {}
class Fish {
    Fish getFish() throws BubbleException {
        throw new RuntimeException("fish!");
    }
}
public final class Clownfish extends Fish {
    public final Clownfish getFish() {
        throw new RuntimeException("clown!");
    }
    public static void main(String[] bubbles) {
        final Fish f = new Clownfish();
        f.getFish();
        System.out.println("swim!");
    }
}

```

1. None. The code compiles and prints swim!.
2. None. The code compiles and prints a stack trace.
3. One
4. Two
5. Three

18. How many lines does this code output?

```

import java.util.*;
import java.util.function.*;

public class PrintNegative {

    public static void main(String[] args) {
        List<Integer> list= new ArrayList<>();
        list.add(-5);
        list.add(0);
        list.add(5);
        print(list, e -> e < 0);
    }

    public static void print(List<Integer> list, Predicate<Integer> p
        for (Integer num : list)
            if (p.test(num))
                System.out.println(num);
    }
}

```

1. One
2. Two
3. Three
4. None. It doesn't compile.
5. None. It throws an exception at runtime.

19. Which keywords are required with a try statement?

1. finalize
 2. catch
 3. throws
 4. finally
1. I only
 2. II only
 3. III only
 4. IV only

- 5. I or II, or both
- 6. None of the above

20. What is the output of the following?

```
12: int result = 8;
13: loop: while (result > 7) {
14:     result++;
15:     do {
16:         result--;
17:     } while (result > 5);
18:     break loop;
19: }
20: System.out.println(result);
```

- 1. 5
- 2. 7
- 3. 8
- 4. The code does not compile.
- 5. The code compiles but throws an exception at runtime.

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

```
package reptile;
public class Alligator {
    static int teeth;
    double scaleToughness;
    public Alligator() {
        teeth++;
    }
    public void snap(int teeth) {
        System.out.print(teeth+" ");
        teeth--;
    }
    public static void main(String[] unused) {
        new Alligator().snap(teeth);
        new Alligator().snap(teeth);
    }
}
```

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

22. What is the output of the following?

```
public class Costume {
    public static void main(String[] black) {
        String witch = "b";
        String tail = "lack";
        witch.concat(tail);
        System.out.println(witch);
    }
}
```

- 1. b
- 2. black
- 3. lack
- 4. The code does not compile.
- 5. The code compiles but throws an exception at runtime.

23. Which modifiers can be independently applied to an interface method? (Choose three.)

- 1. default
- 2. protected
- 3. static
- 4. private
- 5. final
- 6. abstract

24. What is the output of the following?

```
public class Shoelaces {  
    public static void main(String[] args) {  
        String tie = null;  
        while (tie = null)  
            tie = "shoelace";  
        System.out.print(tie);  
    }  
}
```

1. null
2. shoelace
3. shoelaceshoelace
4. The code does not compile.
5. This is an infinite loop.
6. The code compiles but throws an exception at runtime.

25. What statements are true about compiling a Java class file? (Choose two.)

1. If the file does not contain a package statement, then the compiler considers the class part of the `java.lang` package.
2. The compiler assumes every class implicitly imports the `java.lang.*` package.
3. The compiler assumes every class implicitly imports the `java.util.*` package.
4. Java requires every file to declare a package statement.
5. Java requires every file to declare at least one `import` statement.
6. If the class declaration does not extend another class, then it implicitly extends the `java.lang.Object` class.

26. What is the output of the following application?

```
package woods;  
interface Plant {  
    default String grow() { return "Grow!"; }  
}  
interface Living {  
    public default String grow() { return "Growing!"; }  
}  
public class Tree implements Plant, Living { // m1  
    public String grow(int height) { return "Super Growing!"; }  
    public static void main(String[] leaves) {  
        Plant p = new Tree(); // m2  
        System.out.print(((Living)p).grow()); // m3  
    }  
}
```

1. Grow!
2. Growing!
3. Super Growing!
4. It does not compile because of line m1.
5. It does not compile because of line m2.
6. It does not compile because of line m3.

27. What is the result of the following?

```
public static void main(String... args) {  
    String name = "Desiree";  
    int _number = 694;  
    boolean profit$$$;  
    System.out.println(name + " won. "  
        + _number + " profit? " + profit$$$);  
}
```

1. The declaration of `name` does not compile.
2. The declaration of `_number` does not compile.
3. The declaration of `profit$$$` does not compile.
4. The `println` statement does not compile.
5. The code compiles and runs successfully.
6. The code compiles and throws an exception at runtime.

28. Fill in the blanks: Given a variable x, _____ decreases the value of x by 1 and returns the original value, while _____ increases the value of x by 1 and returns the new value.

1. x--, ++x
2. x--, x++
3. --x, x++
4. --x, ++x

29. Given the following two classes in the same package, which constructors contain compiler errors? (Choose three.)

```
public class Big {
    public Big(boolean stillIn) {
        super();
    }
}

public class Trouble extends Big {
    public Trouble() {}
    public Trouble(int deep) {
        super(false);
        this();
    }
    public Trouble(String now, int... deep) {
        this(3);
    }
    public Trouble(long deep) {
        this("check", deep);
    }
    public Trouble(double test) {
        super(test > 5 ? true : false);
    }
}
```

1. public Big(boolean stillIn)
2. public Trouble()
3. public Trouble(int deep)
4. public Trouble(String now, int... deep)
5. public Trouble(long deep)
6. public Trouble(double test)

30. Which of the following can replace the comment so this code outputs 100? (Choose two.)

```
public class Stats {
    // INSERT CODE
    public static void main(String[] math) {
        System.out.println(max - min);
    }
}
```

1. final int min, max = 100;
2. final int min = 0, max = 100;
3. int min, max = 100;
4. int min = 0, max = 100;
5. static int min, max = 100;
6. static int min = 0, max = 100;

31. Which of the following statements are true about Java operators and statements? (Choose two.)

1. Both right-hand sides of the ternary expression will be evaluated at runtime.
2. A switch statement may contain at most one default statement.
3. A single if-then statement can have multiple else statements.
4. The | and || operator are interchangeable, always producing the same results at runtime.
5. The ! operator may not be applied to numeric expressions.

32. What is the output of the following?

```
1: public class Legos {
2:     public static void main(String[] args) {
3:         StringBuilder sb = new StringBuilder();
4:         sb.append("red");
5:     }
6: }
```



```

5:         sb.deleteCharAt(0);
6:         sb.delete(1, 1);
7:         System.out.println(sb);
8:     }
9: }

```

1. r

2. e

3. ed

4. red

5. The code does not compile.

6. The code compiles but throws an exception at runtime.

33. Which of the following is a valid method name in Java? (Choose two.)

1. ____()

2. %run()

3. check-Activity()

4. \$Hum2()

5. sing\\3()

6. po#ut ()

34. Which of the following statements about inheritance are true? (Choose two.)

1. Inheritance is better than using static methods for accessing data in other classes.

2. Inheritance allows a method to be overridden in a subclass, possibly changing the expected behavior of other methods in a superclass.

3. Inheritance allows objects to inherit commonly used attributes and methods.

4. It is possible to create a Java class that does not inherit from any other.

5. Inheritance tends to make applications more complicated.

35. Which of the following statements about Java are true?

1. The java command uses . to separate packages.

2. Java supports functional programming.

3. Java is object oriented.

4. Java supports polymorphism.

1. I only

2. II only

3. II and III

4. I, III, and IV

5. I, II, III, and IV

6. None are true.

36. What is the output of the following?

```

String[][] listing = new String[][] { { "Book", "34.99" },
    { "Game", "29.99" }, { "Pen", ".99" } };
System.out.println(listing.length + " " + listing[0].length);

```

1. 2 2

2. 2 3

3. 3 2

4. 3 3

5. The code does not compile.

6. The code compiles but throws an exception at runtime.

37. Which of the following variable types is permitted in a switch statement? (Choose three.)

1. Character

2. Byte
3. Double
4. long
5. String
6. Object

38. What does the following do?

```
public class Shoot {
    interface Target {
        boolean needToAim(double angle);
    }
    static void prepare(double angle, Target t) {
        boolean ready = t.needToAim(angle); // k1
        System.out.println(ready);
    }
    public static void main(String[] args) {
        prepare(45, d => d > 5 || d < -5); // k2
    }
}
```

1. It prints true.
2. It prints false.
3. It doesn't compile due to line k1.
4. It doesn't compile due to line k2.
5. It doesn't compile due to another line.

39. Which of the following is a valid code comment in Java? (Choose three.)

1. /** Insert */ in next method **/
2. /***** Find the kitty cat */
3. // Is this a bug?
4. / Begin method - performStart() /
5. /** TODO: Call grandma */
6. # Updated code by Patti

40. Given the following two classes, each in a different package, which lines allow the second class to compile when inserted independently? (Choose two.)

```
package food;
public class Grass {
    public static int seeds = 10;
    public static Grass getGrass() {return new Grass();}
}

package woods;
// INSERT CODE HERE
public class Deer {
    public void eat() {
        getGrass();
        System.out.print(seeds);
    }
}
```

1. import static food.Grass.getGrass;
1. import static food.Grass.seeds;
2. import static food.*;
3. static import food.Grass.*;
4. import food.Grass.*;
5. static import food.Grass.getGrass;
1. static import food.Grass.seeds;
6. import static food.Grass.*;

41. What is the result of the following?

```
import java.util.*;
public class Museums {
    public static void main(String[] args) {
        String[] array = {"Natural History", "Science", "Art"};
        List<String> museums = Arrays.asList(array);
        museums.remove(2);
        System.out.println(museums);
    }
}
```

```

    }

```

1. [Natural History, Science]
2. [Natural History, Science, Art]
3. The code does not compile.
4. The code compiles but throws an exception at runtime.

42. Which of the following substitutions will compile? (Choose two.)

```

public class Underscores {
    public String name = "Sherrin";
    public void massage() {
        int zip = 10017;
    }
}

```

1. Change name to _name
2. Change 10017 to _10017
3. Change 10017 to 10017_
4. Change 10017 to 10_0_17
5. Change int to _int

43. What is the result of the following when called as java counting.Binary?

```

package counting;
import java.util.*;
public class Binary {

    public static void main(String[] args) {
        args = new String[] { "0", "1", "01", "10" };

        Arrays.sort(args);
        System.out.println(Arrays.toString(args));
    }
}

```

1. []
2. [0, 01, 1, 10]
3. [0, 01, 10, 1]
4. [0, 1, 01, 10]
5. The code does not compile.
6. The code compiles but throws an exception at runtime.

44. Fill in the blanks: Using the _____ and _____ modifiers together allows a variable to be accessed from any class, without requiring an instance variable.

1. final, package-private
2. class, static
3. protected, instance
4. public, static
5. default, public

45. How many lines does the following code output?

```

import java.util.*;
public class Exams {
    public static void main(String[] args) {
        List<String> exams = Arrays.asList("OCA", "OCP");
        for (String e1 : exams)
            for (String e2 : exams)
                System.out.print(e1 + " " + e2);
                System.out.println();
    }
}

```

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

46. Which of the following are true statements? (Choose two.)

1. The javac command compiles a source text file into a set of machine instructions.
2. The java command compiles a .class file into a .java file.
3. The javac command compiles a .java file into a .class file.
4. The javac command compiles a source text file into a bytecode file.
5. The java command compiles a .java file into a .class file.
6. The javac command compiles a .class file into a .java file.

47. How many of the following lines of code compile?

```
char one = Integer.parseInt("1");
Character two = Integer.parseInt("2");
int three = Integer.parseInt("3");
Integer four = Integer.parseInt("4");
short five = Integer.parseInt("5");
Short six = Integer.parseInt("6");
```

1. None
2. One
3. Two
4. Three
5. Four
6. Five

48. Given the application below, what data types can be inserted into the blank that would allow the code to print 3? (Choose three.)

```
public class Highway {
    public int drive(long car) { return 2; }
    public int drive(double car) { return 3; }
    public int drive(int car) { return 5; }
    public int drive(short car) { return 3; }
    public static void main(String[] gears) {
        _____ value = 5;
        System.out.println(new Highway().drive(value));
    }
}
```

1. boolean
2. short
3. int
4. byte
5. long
6. float

49. How many times does this code print true?

```
import java.time.*;
public class Equality {
    public void main(String[] args) {
        System.out.println(new StringBuilder("zelda")
            == new StringBuilder("zelda"));
        System.out.println(3 == 3);
        System.out.println("bart" == "bart");
        System.out.println(new int[0] == new int[0]);
        System.out.println(LocalTime.now() == LocalTime.now());
    }
}
```

1. None
2. One
3. Two
4. Three
5. The code does not compile.

50. What is the output of the following application?

```
package ballroom;
public class Dance {
    public static void swing(int... beats) throws ClassCastException
    try {
        System.out.print("1"+beats[2]); // p1
    } catch (RuntimeException e) {
```

```

        System.out.print("2");
    } catch (Exception e) {
        System.out.print("3");
    } finally {
        System.out.print("4");
    }
}
public static void main(String... music) {
    new Dance().swing(0,0); // p2
    System.out.print("5");
}
}

```

-
1. 145
 2. 1045
 3. 24, followed by a stack trace
 4. 245
 5. The code does not compile because of line p1.
 6. The code does not compile because of line p2.

51. What is the output of the following?

```

List<String> drinks = Arrays.asList("can", "cup");
for (int container = drinks.size(); container > 0; container++) {
    System.out.print(drinks.get(container-1) + ",");
}

```

-
1. can, cup,
 2. cup, can,
 3. The code does not compile.
 4. This is an infinite loop.
 5. The code compiles but throws an exception at runtime.

52. Which of the following method signatures are valid declarations of an entry point in a Java application? (Choose three.)

1. `public static void main(String... widgets)`
2. `public static void main(String sprockets)`
3. `protected static void main(String[] args)`
4. `public static int void main(String[] arg)`
5. `public static final void main(String []a)`
6. `public static void main(String[] data)`

53. Given the application below and the choices available, which lines must all be removed to allow the code to compile? (Choose three.)

```

1: package year;
2: public class Seasons {
3:     public static void main(String[] time) {
4:         final long winter = 10;
5:         final byte season = 2;
6:         int fall = 4;
7:         final short summer = 3;
8:         switch(season) {
9:             case 1:
10:                case winter: System.out.print("winter");
11:                default:
12:                case fall: System.out.print("fall");
13:                case summer: System.out.print("summer");
14:                default:
15:            }
16:        }
17: }

```

-
1. Line 8
 2. Line 9
 3. Line 10
 4. Line 11
 5. Line 12
 6. Line 13

54. Given the application below, which lines do not compile? (Choose three.)

```
package furryfriends;
interface Friend {
    protected String getName(); // h1
}
class Cat implements Friend {
    String getName() { // h2
        return "Kitty";
    }
}
public class Dog implements Friend {
    String getName() throws RuntimeException { // h3
        return "Doggy";
    }
    public static void main(String[] adoption) {
        Friend friend = new Dog(); // h4
        System.out.print(((Cat)friend).getName()); // h5
        System.out.print(((Dog)null).getName()); // h6
    }
}
```

1. Line h1
2. Line h2
3. Line h3
4. Line h4
5. Line h5
6. Line h6

55. Which of the following are unchecked exceptions? (Choose three.)

1. FileNotFoundException
2. ArithmeticException
3. IOException
4. Exception
5. IllegalArgumentException
6. RuntimeException

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

```
package ranch;
public class Cowboy {
    private int space = 5;
    private double ship = space < 2 ? 1 : 10; // g1
    public void printMessage() {
        if(ship>1) {
            System.out.println("Goodbye");
        } if(ship<10 && space>=2) System.out.println("Hello"); // g2
        else System.out.println("See you again");
    }
    public static final void main(String... stars) {
        new Cowboy().printMessage();
    }
}
```

1. It only prints Hello.
2. It only prints Goodbye.
3. It only prints See you again.
4. It does not compile because of line g1.
5. It does not compile because of line g2.
6. None of the above

57. Given the following three class declarations, which sets of access modifiers can be inserted, in order, into the blank lines below that would allow all of the classes to compile? (Choose three.)

```
package wake;
public class Alarm {
    _____ static int clock;
    _____ long getTime() {return clock;}
}

package wake;
public class Coffee {
    private boolean bringCoffee() { return new Alarm().clock<10;}
}
```

```
package sleep;
public class Snooze extends wake.Alarm {
    private boolean checkTime() { return getTime()>10;}
}
```

-
1. protected and package-private (blank)
 2. public and public
 3. package-private (blank) and protected
 4. protected and protected
 5. private and public
 6. package-private (blank) and package-private (blank)

58. Given that `FileNotFoundException` is a subclass of `IOException` and `Long` is a subclass of `Number`, what is the output of the following application?

```
package materials;

import java.io.*;

class CarbonStructure {
    protected long count;
    public abstract Number getCount() throws IOException; // q1
    public CarbonStructure(int count) { this.count = count; }
}

public class Diamond extends CarbonStructure {
    public Diamond() { super(15); }
    public Long getCount() throws FileNotFoundException { // q2
        return count;
    }
    public static void main(String[] cost) {
        try {
            final CarbonStructure ring = new Diamond(); // q3
            System.out.print(ring.getCount()); // q4
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

-
1. 15
 2. It does not compile because of line q1.
 3. It does not compile because of line q2.
 4. It does not compile because of line q3.
 5. It does not compile because of line q4.
 6. It compiles but throws an exception at runtime.

59. How many lines contain a compile error?

```
1: import java.time.*;
2: import java.time.format.*;
3:
4: public class HowLong {
5:     public void main(String h) {
6:         LocalDate newYears = new LocalDate(2017, 1, 1);
7:         Period period = Period.ofYears(1).ofDays(1);
8:         DateTimeFormat format = DateTimeFormat.ofPattern("MM-dd-");
9:         System.out.print(format.format(newYears.minus(period)));
10:    }
11: }
```

-
1. None
 2. One
 3. Two
 4. Three
 5. Four
 6. Five

60. Which of the following statements about try-catch blocks are correct? (Choose two.)

1. A catch block can never appear after a finally block.

2. A try block must be followed by a catch block.
3. A finally block can never appear after a catch block.
4. A try block must be followed by a finally block.
5. A try block can have zero or more catch blocks.
6. A try block can have zero or more finally blocks.

61. What is printed by the following code snippet?

```
int fish = 1 + 2 * 5 >= 2 ? 4 : 2;
int mammals = 3 < 3 ? 1 : 5 >= 5 ? 9 : 7;
System.out.print(fish+mammals+"");
```

1. 49
2. 13
3. 18
4. 99
5. It does not compile.

62. Which of the following statements about objects, reference types, and casting are correct? (Choose three.)

1. An object can be assigned to an inherited interface reference variable without an explicit cast.
2. The compiler can prevent all explicit casts that lead to an exception at runtime.
3. Casting an object to a reference variable does not modify the object in memory.
4. An object can be assigned to a subclass reference variable without an explicit cast.
5. An object can be assigned to a superclass reference variable without an explicit cast.
6. An implicit cast of an object to one of its inherited types can sometimes lead to a `ClassCastException` at runtime.

63. What is the output of the following when run as `java EchoFirst seed flower plant`?

```
package unix;

import java.util.*;

public class EchoFirst {

    public static void main(String[] args) {
        int result = Arrays.binarySearch(args, args[0]);
        System.out.println(result);
    }
}
```

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

64. How many objects are eligible for garbage collection at the end of the `main()` method?

```
package store;
public class Shoes {

    static String shoe1 = new String("sandal");
    static String shoe2 = new String("flip flop");

    public static void shopping() {
        String shoe3 = new String("croc");
        shoe2 = shoe1;
        shoe1 = shoe3;
    }

    public static void main(String... args) {
        shopping();
    }
}
```

 }

1. None
2. One
3. Two
4. Three
5. The code does not compile.

65. Fill in the blanks: The _____ keyword is used in method declarations, the _____ keyword is used to guarantee a statement will execute even if an exception is thrown, and the _____ keyword is used to throw an exception to the surrounding process.

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

66. Which statements best describe the result of this code? (Choose two.)

```
package nyc;
public class TouristBus {
    public static void main(String... args) {
        String[] nycTourLoops = new String[] { "Downtown", "Uptown", "I
        String[] times = new String[] { "Day", "Night" };
        for (int i = 0, j = 0; i < nycTourLoops.length; i++, j++)
            System.out.println(nycTourLoops[i] + " " + times[j]);
    }
}
```

1. The println causes one line of output.
2. The println causes two lines of output.
3. The println causes three lines of output.
4. The code terminates successfully.
5. The code throws an exception at runtime.

67. Fill in the blanks: Because of _____, it is possible to _____ a method, which allows Java to support _____.

1. abstract methods, override, inheritance
2. concrete methods, overload, inheritance
3. virtual methods, overload, interfaces
4. inheritance, abstract, polymorphism
5. virtual methods, override, polymorphism.

68. What is the result of the following?

```
package calendar;
public class Seasons {

    public static void seasons(String... names) {
        int l = names[1].length();    // s1
        System.out.println(names[1]); // s2
    }

    public static void main(String[] args) {
        seasons("Summer", "Fall", "Winter", "Spring");
    }
}
```

1. Fall
2. Spring
3. The code does not compile.
4. The code throws an exception on line s1.
5. The code throws an exception on line s2.

69. How many lines of the following application contain compilation errors?

```
1: package percussion;
2:
3: interface MakesNoise {}
4: abstract class Instrument implements MakesNoise {
5:     public Instrument(int beats) {}
6:     public void play() {}
7: }
8: public class Drum extends Instrument {
9:     public void play(int count) {}
10:    public void concert() {
11:        super.play(5);
12:    }
13:    public static void main(String[] beats) {
14:        MakesNoise mn = new Drum();
15:        mn.concert();
16:    }
17: }
```

1. None. The code compiles and runs without issue.
2. One
3. Two
4. Three
5. Four

70. What is the output of the following application?

```
package fly;
public class Helicopter {
    public int adjustPropellers(int length, String[] type) {
        length++;
        type[0] = "LONG";
        return length;
    }
    public static void main(String[] climb) {
        final Helicopter h = new Helicopter();
        int length = 5;
        String[] type = new String[1];
        length = h.adjustPropellers(length, type);
        System.out.print(length+" "+type[0]);
    }
}
```

1. 5, LONG
2. 6, LONG
3. 5, null
4. 6, null
5. The code does not compile.
6. The code compiles but throws an exception at runtime.

71. How many lines of the following application do not compile?

```
package castles;
class OpenDoorException extends Exception {}
class CableSnapException extends OpenDoorException {}
public class Palace {
    public void openDrawbridge() throws Exception {
        try {
            throw new Exception("Problem");
        } catch (OpenDoorException e) {
            throw new OpenDoorException();
        } catch (CableSnapException ex) {
            try {
                throw new OpenDoorException();
            } catch (Exception ex) {
            } finally {
                System.out.println("Almost done");
            }
        } finally {
            throw new RuntimeException("Unending problem");
        }
    }
    public static void main(String[] moat) throws IllegalArgumentException {
        new Palace().openDrawbridge();
    }
}
```

1. None. The code compiles and produces a stack trace at runtime.
2. One
3. Two
4. Three

- 5. Four
- 6. Five

72. Choose the best answer: _____ and _____ are two properties that go hand in hand to improve class design by structuring a class with related attributes and actions while protecting the underlying data from access by other classes.

- 1. Optimization and platform independence
- 2. Platform independence and encapsulation
- 3. Platform independence and inheritance
- 4. Object orientation and encapsulation
- 5. Inheritance and polymorphism

73. What is the output of the following?

```
string bike1 = "speedy";
string bike2 = new String("speedy");
boolean test1 = bike1 == bike2;
boolean test2 = bike1.equals(bike2);
System.out.println(test1 + " " + test2);
```

- 1. false false
- 2. false true
- 3. true false
- 4. true true
- 5. The code does not compile.
- 6. The code compiles but throws an exception at runtime.

74. What is the output of the following when run as `java EchoFirst seed f1ower plant?`

```
package unix;

import java.util.*;

public class EchoFirst {

    public static void main(String[] args) {
        Arrays.sort(args);
        int result = Arrays.binarySearch(args, args[0]);
        System.out.println(result);
    }
}
```

- 1. 0
- 2. 1
- 3. 2
- 4. The code does not compile.
- 5. The code compiles but throws an exception at runtime.
- 6. The output is not guaranteed.

75. Which are true statements? (Choose three.)

- 1. Every `do-while` loop can be rewritten as a `for-each` loop.
- 2. Every `for-each` loop can be rewritten as a `do-while` loop.
- 3. Every `for-each` loop can be rewritten as a traditional `for` loop.
- 4. Every `for-each` loop can be rewritten as a `while` loop.
- 5. Every traditional `for` loop can be rewritten as a `for-each` loop.
- 6. Every `while` loop can be rewritten as a `for-each` loop.

76. How many lines does this program print?

```
import java.time.*;

public class OnePlusOne {
    public static void main(String... nums) {
        LocalDate time = LocalDate.of(1, 11);
        while (time.getHour() < 1) {
            time.plusHours(1);
            System.out.println("in loop");
        }
    }
}
```

```
    }
}
```

1. None
2. One
3. Two
4. This is an infinite loop.
5. The code does not compile.

77. How many objects are eligible for garbage collection immediately before the end of the `main()` method?

```
public class Tennis {
    public static void main(String[] game) {
        String[] balls = new String[1];
        int[] scores = new int[1];
        balls = null;
        scores = null;
    }
}
```

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

78. What is the output of the following?

```
14: int count = 0;
15: LocalDate date = LocalDate.of(2017, Month.JANUARY, 1);
16: while (date.getMonth() != Month.APRIL)
17:     date = date.minusMonths(1);
18:     count++;
19: System.out.println(count);
```

1. 0
2. 1
3. 3
4. 9
5. This is an infinite loop.
6. The code does not compile.

79. How many lines of the following class do not compile?

```
1: package arctic;
2: abstract class Bear {
3:     protected int sing;
4:     protected abstract int grunt();
5:     int sing() {
6:         return sing;
7:     }
8: }
9: public class PolarBear extends Bear {
10:     int grunt() {
11:         sing() += 10;
12:         return super.grunt()+1;
13:     }
14: }
15: }
```

1. None, the class compiles without issue.
2. One
3. Two
4. Three
5. Four
6. Five

80. In which places is the `default` keyword permitted to be used? (Choose two.)

1. Access modifier in a class
2. Execution path in a `switch` statement
3. Method name

- 4. Modifier in an abstract interface method
- 5. Modifier in an interface method with a body
- 6. Variable name



◀ PREV
Chapter 9 Working with Selected Classes from the Java API

NEXT ▶
Part II OCP