This chapter contains 50 questions and is designed to simulate a real 1Z0-819 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 90 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 result of executing the following method?

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

A. The declaration of `name` does not compile.

B. The declaration of `_number` does not compile.

C. The declaration of `profit$$$` does not compile.

D. The `println()` statement does not compile.

E. The code compiles and runs successfully.

F. The code compiles and throws an exception at runtime.

2. Which statements about try-with-resources are true? (Choose two.)

A. Any resource used must implement `Closeable`.

B. If more than one resource is used, then the order in which they are closed is the reverse of the order in which they were created.

C. If the `try` block and `close()` method both throw an exception, the one thrown by the `try` block is suppressed.

D. Neither a `catch` nor a `finally` block is required.

E. The `close()` method of the resources must throw a checked exception.

3. Bill wants to create a program that reads all of the lines of all of his books using NIO.2. Unfortunately, Bill may have made a few mistakes writing his program. How many lines of the following class contain compilation errors?

```
1:  package bookworm;
2:  import java.io.*;
```

```
3:  import java.nio.file.*;
4:  public class ReadEverything {
5:     public void readFile(Path p) {
6:        try {
7:           Files.readAllLines(p)
8:           .parallel()
9:           .forEach(System.out::println);
10:       } catch (Exception e) {}
11:    }
12:    public void read(Path directory) throws Exception {
13:       Files.walk(directory)
14:          .filter(p -> File.isRegularFile(p))
15:          .forEach(x -> readFile(x));
16:    }
17:    public static void main(String[] b) throws IOException {
18:       Path p = Path.get("collection");
19:       new ReadEverything().read(p);
20:    }
21: }
```

A. None. Bill's implementation is correct.
B. One.
C. Two.
D. Three.
E. Four.
F. Five.

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

A. `/** Insert */ in next method **/`
B. `/****** Find the kitty cat */`
C. `// Is this a bug?`
D. `$ Begin method - performStart() $`
E. `/*** TODO: Call grandma ***/`
F. `# Updated code by Patti`

5. What is the minimum number of `requires` directives that need to be removed to break the cyclic dependency?

```
module com.animal {
   exports com.animal;
   requires com.plant;
}
```

```
module com.plant {
   exports com.plant;
   requires com.animal;
}
module com.worm {
   exports com.worm;
   requires com.animal;
   requires com.plant;
}
module com.hedgehog {
   exports com.hedgehog;
   requires com.animal;
   requires com.plant;
}
```

A. None, there is no cyclic dependency

B. 1

C. 2

D. 3

E. 4

6. What is the result of the following?

```
package calendar;
public class Seasons {
   public static void seasons(String... names) {
      var v = names[1].length();       // s1
      System.out.println(names[v]);    // s2
   }
   public static void main(String[] args) {
      seasons("Summer", "Fall", "Winter", "Spring");
   }
}
```

A. `Fall`

B. `Spring`

C. The code does not compile.

D. The code throws an exception on line `s1`.

E. The code throws an exception on line `s2`.

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

```
import java.util.*;

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

   A. `0`
   B. `1`
   C. `2`
   D. The code does not compile.
   E. The code compiles but throws an exception at runtime.
   F. The output is not guaranteed.

8. Which method can fill in the blank that would cause the program to consistently print `Tie!` ten times?

```
import java.util.concurrent.*;
import java.util.concurrent.locks.*;
public class TieShoes {
   private Lock shoes = new ReentrantLock();
   public void tie() {
      try {
         if (shoes._________________) {
            System.out.println("Tie!");
            shoes.unlock();
         }
      } catch (Exception e) {}
   }
   public static void main(String... unused) {
      var gate = new TieShoes();
      for (int i = 0; i < 10; i++) {
         new Thread(() -> gate.tie()).start();
      }
   } }
```

   A. `lock()`
   B. `tryLock()`

C. `tryLock(10)`

D. The code does not compile regardless of what is placed in the blank.

E. None of the above.

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

```
// Alarm.java
package wake;
public class Alarm {
   _______ static int clock;
   _______ long getTime() {return clock;}
}
// Coffee.java
package wake;
public class Coffee {
   private boolean bringCoffee() {
      return new Alarm().clock<10;
   }
}

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

A. package-private (blank) and package-private (blank)

B. package-private (blank) and `protected`

C. `protected` and package-private (blank)

D. `protected` and `protected`

E. `private` and `public`

F. `public` and `public`

10. What is the result of the following?

```
var dice = new TreeSet<Integer>();
dice.add(6);
dice.add(6);
dice.add(4);
```

```
dice.stream()
   .filter(n -> n != 4)
   .forEach(System.out::println)
   .count();
```

A. It prints just one line.

B. It prints one line and then the number 3.

C. There is no output.

D. The code does not compile.

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

11. Suppose the `pandas` table has one row with the name `Mei Xiang` and the location `DC`. What does the following code output?

```
var url = "jdbc:derby:pandas";
var sql = "SELECT name FROM pandas WHERE location = ?";
try (var conn = DriverManager.getConnection(url);
   var stmt = conn.prepareStatement(sql);          // s1
   stmt.setString(1, "DC");
   var rs = stmt.executeQuery()) {

   if (rs.next())
      System.out.println(rs.getString("name"));   // s2
   else
      System.out.println("No match");
}
```

A. `Mei Xiang`

B. `No match`

C. The code does not compile due to line `s1`.

D. The code does not compile due to line `s2`.

E. The code does not compile due to another line.

F. The code throws an exception at runtime.

12. Which of the following can fill in the blank to print out just the number `161`?

```
import java.util.*;
import java.util.stream.*;
class Runner {
   private int numberMinutes;
   public Runner(int n) {
      numberMinutes = n;
   }
```

```
   public int getNumberMinutes() {
      return numberMinutes;
   } }
public class Marathon {
   public static void main(String[] args) {
      var match = Optional.ofNullable(161);    // line z
      var runners = Stream.of(new Runner(183),
         new Runner(161), new Runner(201));
      var opt = runners_____________________;
   } }
```

A.

```
.map(Runner::getNumberMinutes)
.filter(m -> match.get().equals(m))
.peek(System.out::println)
.count()
```

B.

```
.map(Runner::getNumberMinutes)
.filter(m -> match.get().equals(m))
.peek(System.out::println)
.max()
```

C.

```
.map(Runner::getNumberMinutes)
.peek(System.out::println)
.filter(m -> match.get().equals(m))
.count()
```

D.

```
.map(Runner::getNumberMinutes)
.peek(System.out::println)
.filter(m -> match.get().equals(m))
.max()
```

E. The code does not compile due to line z.

F. None of the above.

13. Which of the following exceptions do not need to be handled or declared by the method in which they are thrown? (Choose three.)

A. `FileNotFoundException`

B. `ArithmeticException`

C. `IOException`

D. `BigProblem`

E. `IllegalArgumentException`

F. `RuntimeException`

14. What is the output of the following application?

```
package homework;
import java.util.*;
import java.util.stream.*;
public class QuickSolution {
   public static int findFast(Stream<Integer> s) {
      return s.findAny().get();
   }
   public static int findSlow(Stream<Integer> s) {
      return s.parallel().findFirst().get();
   }

   public static void main(String[] pencil) {
      var s1 = List.of(1,2,3,4,5).stream();
      var s2 = List.of(1,2,3,4,5).stream();
      int val1 = findFast(s1);
      int val2 = findSlow(s2);
      System.out.print(val1+" "+val2);
   }
}
```

A. `1 1`

B. `3 1`

C. The answer cannot be determined until runtime.

D. The code does not compile.

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

F. None of the above.

15. What is the result of the following?

```
import java.time.*;
import java.time.format.*;
public class PiDay {
   public static void main(String[] args) {
      LocalDateTime pi = LocalDateTime.of(2017, 3, 14, 1, 59);
      DateTimeFormatter formatter = DateTimeFormatter
         .ofPattern("m.ddhh'MM'");
      System.out.print(formatter.format(pi));
   }
}
```

A. `3.011459`

B. `3.1401MM`

C. `59.011459`

D. `59.1401MM`

E. The code does not compile.

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

16. Which is part of the module service and has a `requires` directive?

A. Consumer

B. Service locator

C. Service provider

D. Service provider interface

E. None of the above

17. What option names are equivalent to `-p` and `-cp` on the `javac` command? (Choose two.)

A. `--module-path` and `-classpath`

B. `--module-path` and `-class-path`

C. `--module-path` and `--class-path`

D. `--path` and `-classpath`

E. `--path` and `-class-path`

F. `--path` and `--class-path`

18. What is the result of the following when called as `java Binary.java`?

```
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));
   }
}
```

A. `[]`

B. `[0, 01, 1, 10]`

C. `[0, 01, 10, 1]`

D. `[0, 1, 01, 10]`

E. The code does not compile.

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

19. What is the output of the following application?

```
package music;
interface DoubleBass {
   void strum();
```

```
   default int getVolume() {return 5;}
}
interface BassGuitar {
   void strum();
   default int getVolume() {return 10;}
}
abstract class ElectricBass implements DoubleBass, BassGuitar {
   @Override public void strum() {System.out.print("X");}
}
public class RockBand {
   public static void main(String[] strings) {
      final class MyElectricBass extends ElectricBass {
         public int getVolume() {return 30;}
         public void strum() {System.out.print("Y");}
      }
   } }
```

A. `X`

B. `Y`

C. The application completes without printing anything.

D. `ElectricBass` is the first class to not compile.

E. `RockBand` is the first class to not compile.

F. None of the above.

20. 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
   }
}
```

A. It prints `true`.

B. It prints `false`.

C. It doesn't compile due to line `k1`.

D. It doesn't compile due to line `k2`.

E. It doesn't compile due to another line.

21. Which of the following lambda expressions can be passed to a method that takes `IntUnaryOperator` as an argument? (Choose three.)

A. `v -> {System.out.print("Hello!"); return 2%1;}`

B. `(Integer w) -> w.intValue()`

C. `(int j) -> (int) 30L`

D. `(int q) -> q / 3.1`

E. `(long x) -> (int) x`

F. `z -> z`

22. How many of these module declarations are valid?

```
module com.apple { exports com.apple; }
module com.4apple { requires com.apple;}
module com.apple4 { declares com.apple; }
module com.apple-four { }
module com.apple$ {}
```

A. None.

B. One.

C. Two.

D. Three.

E. Four.

F. Five.

23. What is the output of the following application?

```
package tax;
public class Accountant {
   public void doTaxes() throws Throwable {
      try {
         throw new NumberFormatException();
      } catch (ClassCastException
            | ArithmeticException e) { // p1
         System.out.println("Math");
      } catch (IllegalArgumentException | Exception f) { // p2
         System.out.println("Unknown");
      }
   }
   public static void main(String[] numbers) throws Throwable {
      try {
         new Accountant().doTaxes();
      } finally {
         System.out.println("Done!");
      }
   }
}
```

A. `Math`

B. `Unknown`

C. `Unknown` followed by `Done!`

D. The code does not compile due to line `p1`.

E. The code does not compile due to line `p2`.

F. None of the above.

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

```
package names;
import java.util.*;
import java.util.function.*;
interface ApplyFilter {
   void filter(List<String> input);
}
public class FilterBobs {
   static Function<String,String> first = s ->
      {System.out.println(s); return s;};
   static Predicate second = t -> "bob".equalsIgnoreCase(t);
   public void process(ApplyFilter a, List<String> list) {
      a.filter(list);
   }
   public static void main(String[] contestants) {
      final List<String> people = new ArrayList<>();
      people.add("Bob");
      people.add("bob");
      people.add("Jennifer");
      people.add("Samantha");
      final FilterBobs f = new FilterBobs();
      f.process(q -> {
         q.removeIf(second);
         q.forEach(first);
      }, people);
   }
}
```

A. It prints two lines.

B. It prints three lines.

C. One line of code does not compile.

D. Two lines of code do not compile.

E. Three lines of code do not compile.

F. The code compiles but prints an exception at runtime.

25. How many of these variables are `true`?

```
var lol = "lol";
var smiley = lol.toUpperCase() == lol;
var smirk = lol.toUpperCase() == lol.toUpperCase();
var blush = lol.toUpperCase().equals(lol);
var cool = lol.toUpperCase().equals(lol.toUpperCase());
var wink = lol.toUpperCase().equalsIgnoreCase(lol);
var yawn = lol.toUpperCase().equalsIgnoreCase(
   lol.toUpperCase());
```

A. One.

B. Two.

C. Three.

D. Four.

E. Five.

F. None. The code does not compile.

26. Let's say you are managing animals at a veterinary hospital using a new software application. Which metadata attributes would be best managed with an annotation? (Choose two.)

A. The number of animals that are checked at any given time

B. The maximum number of the animals the hospital can hold

C. The feeding schedule for each animal checked in

D. The name of every veterinarian in the building

E. Whether or not the hospital is capable of handling emergencies

F. The location of each animal within the hospital

27. What is the output of the following program?

```
public class Ghost {
   private final String name;
   public Ghost() {
      this(null);
      this.name = "Casper";
   }
   public Ghost(String n) {
      name = "Boo";
   }
   public static void main(String[] sound) {
      var d = new Ghost("Space");
      System.out.println(d.name);
   }
}
```

A. `Casper`

B. `Boo`

C. `Space`

D. The code does not compile.

E. The answer cannot be determined with the information given.

F. None of the above.

28. How many of the following variable declarations compile?

```
1:  import java.util.*;
2:  public class ListOfList {
3:     public void create() {
4:        List<?> n = new ArrayList<>();
5:        List<? extends RuntimeException> o
6:           = new ArrayList<Exception>();
7:        List<? super RuntimeException> p
8:           = new ArrayList<Exception>();
9:        List<T> q = new ArrayList<?>();
10:       List<T extends RuntimeException> r
11:          = new ArrayList<Exception>();
12:       List<T super RuntimeException> s
13:          = new ArrayList<Exception>();
14:    } }
```

A. None.

B. One.

C. Two.

D. Three.

E. Four.

F. Five.

29. 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 var h = new Helicopter();
      var length = 5;
```

```
      var type = new String[1];
      length = h.adjustPropellers(length, type);
      System.out.print(length+","+type[0]);
   }
}
```

A. `5,LONG`
B. `6,LONG`
C. `5,null`
D. `6,null`
E. The code does not compile.
F. The code compiles but throws an exception at runtime.

30. Fill in the blank with code that belongs in a service provider.

```
String cheese = ServiceLoader.load(Mouse.class)
   .stream ()
   .map(_____________)
   .map(Mouse::favoriteFood)
   .findFirst()
   .orElse("");
```

A. `Mouse.get()`
B. `Mouse::get`
C. `Provider.get()`
D. `Provider::get`
E. None of the above

31. Which lines can fill in the blank that would allow the code to compile? (Choose two.)

```
abstract public class Exam {
   boolean pass;
   protected abstract boolean passed();
   class JavaProgrammerCert extends Exam {
      private Exam part1;
      private Exam part2;
      ____________________________________
   }
}
```

A. `boolean passed() { return part1.pass && part2.pass; }`
B. `boolean passed() { return part1.passed() && part2.passed(); }`
C. `private boolean passed() { return super.passed(); }`
D. `public boolean passed() { return part1.passed() && part2.passed(); }`

E. `public boolean passed() { return part1.pass && part2.pass; }`

F. `public boolean passed() { return super.passed(); }`

32. Which of the following are valid lambda expressions? (Choose three.)

A. `() -> {}`

B. `(Double adder) -> {int y; System.out.print(adder); return adder;}`

C. `(Long w) -> {Long w=5; return 5;}`

D. `(int count, vote) -> count*vote`

E. `dog -> dog`

F. `name -> {name.toUpperCase()}`

33. 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: }
```

A. The code compiles and runs without issue.

B. One.

C. Two.

D. Three.

E. Four.

F. None of the above.

34. Given the `Electricity` annotation, how many lines of the `Solar` class contain a compiler error?

```
1:  import java.lang.annotation.*;
2:  @Target(ElementType.METHOD)
3:  public @interface Electricity {
```

```
4:     int[] value() default 100;
5:     short type() default 1;
6:  }
7:  @Electricity() class Solar {
8:     @Electricity(2) @Electricity(0) void charge() {}
9:     @Electricity(value=9) void turnOn() {}
10:    @Electricity(6,5) void install() {}
11:    @Electricity(value=1,7) void turnOff() {}
12:    @Electricity(value=8) void storePower() {}
13: }
```

A. One.

B. Two.

C. Three.

D. Four.

E. Five.

F. Six.

G. None of the above.

35. What is the output of the following application?

```
1:  interface HasHue {String getHue();}
2:  enum COLORS implements HasHue {
3:     red {
4:        public String getHue() {return "FF0000";}
5:     }, green {
6:        public String getHue() {return "00FF00";}
7:     }, blue {
8:        public String getHue() {return "0000FF";}
9:     }
10:    private COLORS() {}
11: }
12: class Book {
13:    static void main(String[] pencils) {}
14: }
15: final public class ColoringBook extends Book {
16:    final void paint(COLORS c) {
17:       System.out.print("Painting: "+c.getHue());
18:    }
19:    final public static void main(String[] crayons) {
20:       new ColoringBook().paint(green);
```

```
21:    }
22: }
```

A. `Painting: 00FF00`

B. One line of code does not compile.

C. Two lines of code do not compile.

D. Three lines of code do not compile.

E. The code compiles but prints an exception at runtime.

F. None of the above.

36. What is the output of the following code snippet?

```
11: Path x = Paths.get(".","song","..","/note");
12: Path y = Paths.get("/dance/move.txt");
13: x.normalize();
14: System.out.println(x.resolve(y));
15: System.out.println(y.resolve(x));
```

A.

```
/./song/../note/dance/move.txt
/dance/move.txt
```

B.

```
/dance/move.txt
/dance/move.txt/note
```

C.

```
/dance/move.txt
/dance/move.txt/./song/../note
```

D.

```
/note/dance/move.txt
../dance/move.txt/song
```

E. The code does not compile.

F. The code compiles but an exception is thrown at runtime.

37. 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: }
```

A. 6
B. X
C. The code does not compile.
D. The code compiles but throws a `NullPointerException` at runtime.
E. The code compiles but throws a different exception at runtime.
F. The output is not guaranteed.

38. What is the output of the following?

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

A. 2 2
B. 2 3
C. 3 2
D. 3 3
E. The code does not compile.
F. The code compiles but throws an exception at runtime.

39. Which of the following are JDBC interfaces in the `java.sql` package?

A. `Driver`, `Query`
B. `Driver`, `ResultSet`
C. `DriverManager`, `Query`
D. `DriverManager`, `ResultSet`
E. `Driver`, `DriverManager`, `Query`
F. `Driver`, `DriverManager`, `ResultSet`

40. Given the following class, which statement is correct?

```
1:  import java.security.*;
2:  import java.util.*;
3:
4:  public class SecretFile {
5:     private String secret;
6:     // Constructors/Getters Omitted
7:
```

```
8:     private static class Folder {
9:        private final SecretFile value;
10:       private final Permission permission;
11:
12:       // Constructors/Getters Omitted
13:    }
14:
15:    public static Permission getPermission(String check) {
16:       // Implementation Omitted
17:    }
18:
19:    private static Map<String, Folder> c = new HashMap<>();
20:    public static SecretFile getSecret(String t) {
21:       var securityRecord = c.get(t);
22:       if (securityRecord != null) {
23:          return securityRecord.getValue();
24:       }
25:
26:       var p = getPermission(t);
27:       AccessController.checkPermission(p);
28:       var pc = p.newPermissionCollection();
29:       pc.add(p);
30:       var secret = AccessController.doPrivileged(
31:          new PrivilegedAction<SecretFile>() {
32:             public SecretFile run() {
33:                return
34:                   new SecretFile(System.getProperty(t));
35:             }},
36:          new AccessControlContext(new ProtectionDomain[] {
37:             new ProtectionDomain(null, pc) }));
38:       c.put(t, new Folder(secret, p));
39:       return secret;
40:    } }
```

A. The class does not contain any security issues.

B. The class contains exactly one security issue.

C. The class contains exactly two security issues.

D. The class contains exactly three security issues.

E. None of the above.

41. 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;
   }
}
```

A. None.
B. One.
C. Two.
D. Three.
E. The code does not compile.
F. None of the above.

42. What is the output of the following application?

```
package rope;
import java.util.concurrent.*;
public class Jump {
   private static void await(CyclicBarrier b) {
      try { b.await(); } catch (Exception e) {}
   }
   public static void main(String[] chalk) {
      ExecutorService s = Executors.newFixedThreadPool(4);
      final var b = new CyclicBarrier(4,
         () -> System.out.print("Jump!"));
      for(int i=0; i<10; i++)
         s.execute(() ->await(b));
      s.shutdown();
   } }
```

A. `Jump!` is printed once, and the program exits.
B. `Jump!` is printed twice, and the program exits.
C. The code does not compile.
D. The output cannot be determined ahead of time.
E. A deadlock is produced at runtime.
F. None of the above.

43. Given the application shown here, 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
   }
}
```

A. Line `h1`
B. Line `h2`
C. Line `h3`
D. Line `h4`
E. Line `h5`
F. Line `h6`

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

A. `class`, `static`
B. `default`, `public`
C. `final`, package-private
D. `protected`, `instance`
E. `public`, `static`
F. None of the above

45. Which statements when inserted independently will throw an exception at runtime? (Choose two.)

```
var x = new LinkedList<>();
x.offer(18);
// INSERT CODE HERE
```

A. `x.peek(); x.peek();`

B. `x.poll(); x.poll();`

C. `x.pop(); x.pop();`

D. `x.remove(); x.remove();`

46. Which of the following shows a valid `Locale` format? (Choose two.)

A. `iw`

B. `UA`

C. `it_ch`

D. `JA_JP`

E. `th_TH`

F. `ES_HN`

47. Which sets of lines can be removed 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:    {
20:       race += "x";
21:    }
22:    while (race.length() <= 4);
23: } while (race.length() < 4);
24: System.out.println(race);
```

A. Lines 15 and 17

B. Lines 16 and 23

C. Lines 17, 18, and 22

D. Line 17

E. Line 22

F. Line 23

48. 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 void shopping() {
      String shoe3 = new String("croc");
      shoe2 = shoe1;
      shoe1 = shoe3;
   }
   public static void main(String... args) {
      new Shoes().shopping();
   }
}
```

A. None.
B. One.
C. Two.
D. Three.
E. The code does not compile.
F. None of the above.

49. Which of the following variable types can be used in a `switch` statement under some circumstances? (Choose three.)

A. An enumerated type
B. `StringBuilder`
C. `Byte`
D. `Double`
E. `var`
F. `Exception`

50. What is the output of the following?

```
1:  import static java.util.stream.Collectors.*;
2:  import java.util.*;
3:
4:  public class Goat {
5:     private String food;
6:
7:     // constructor, getter and toString
```

```
8:
9:     public static void main(String[] args) {
10:       var goats = List.of(
11:          new Goat("can"),
12:          new Goat("hay"),
13:          new Goat("shorts"),
14:          new Goat("hay"));
15:
16:       goats.stream()
17:          .collect(groupingBy(Goat::getFood))
18:          .entrySet()
19:          .stream()
20:          .filter(e -> e.getValue().size() == 2)
21:          .map(e -> e.getKey())
22:          .collect(partitioningBy(e -> e.isEmpty()))
23:          .get(false)
24:          .stream()
25:          .sorted()
26:          .forEach(System.out::print);
27:    }
28: }
```

A. `canshorts`

B. `hay`

C. `hayhay`

D. `shortscan`

E. The code does not compile.

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