This is the third, and final practice exam chapter in the book. Make sure you have 90 minutes and scratch paper before you start. Good luck on both this chapter and the real exam!

1. What is the result of the following code?

```
// Hopper.java
package com.animals;

public class Hopper {
   protected void hop() {
      System.out.println("hop");
   }
}

// Grasshopper.java
package com.insect;
import com.animals.Hopper;

public class Grasshopper extends Hopper {
   public void move() {
      hop();  // p1
   }
}

// HopCounter.java
package com.animals;

public class HopCounter {

   public static void main(String[] args) {
      var hopper = new Grasshopper();
      hopper.move();  // p2
      hopper.hop();   // p3
   }
}
```

A. The code prints `hop` once.

B. The code prints `hop` twice.

C. The first compiler error is on line `p1`.

D. The first compiler error is on line `p2`.

E. The first compiler error is on line `p3`.

2. Which of the following statements about `try/catch` blocks are correct? (Choose two.)
   - **A.** A `catch` block can never appear after a `finally` block.
   - **B.** A `try` block must be followed by a `catch` block.
   - **C.** A `finally` block can never appear after a `catch` block.
   - **D.** A `try` block must be followed by a `finally` block.
   - **E.** A `try` block can have zero or more `catch` blocks.
   - **F.** A `try` block can have zero or more `finally` blocks.

3. Which statements are correct? (Choose two.)
   - **A.** A `Comparable` implementation is often implemented by a lambda.
   - **B.** A `Comparable` object has a `compare()` method.
   - **C.** The `compare()` and `compareTo()` methods have the same contract for the return value.
   - **D.** It is possible to sort the same `List` using different `Comparator` implementations.
   - **E.** Two objects that return `true` for `equals()` will always return `0` when passed to `compareTo()`.

4. How many lines does this code output?

```
import java.util.*;
public class PrintNegative {
  public static void main(String[] args) {
     List<Integer> list = new ArrayList<>();
     list.add(-5);
     list.add(0);
     list.add(5);
     list.removeIf(e -> e < 0);
     list.forEach(x -> System.out.println(x));
  }
}
```

   - **A.** One.
   - **B.** Two.
   - **C.** Three.
   - **D.** None. It doesn't compile.
   - **E.** None. It throws an exception at runtime.

5. How many lines need to be changed to make this code compile?

```
1: public class Massage {
2:    var name = "Sherrin";
3:    public void massage(var num) {
4:       var zip = 10017;
```

```
5:        var underscores = 1_001_7;
6:        var _ = "";
7:    }
8: }
```

A. Zero
B. One
C. Two
D. Three
E. Four

6. Which two conditions best describe one or more threads that appear to be active but are perpetually stuck and never able to finish their task? (Choose two.)

A. Deadlock
B. Livelock
C. Loss of precision
D. Out of memory error
E. Race condition
F. Starvation

7. How many lines of the following interface do not compile?

```
15: public interface Piano {
16:    String type = "Grand";
17:    void play();
18:    public static int getNumberOfKeys() {
19:       return type.equals("Grand") ? 88 : 61;
20:    }
21:    private static void printPianoInfo() {
22:       play();
23:       System.out.println("Key Count: "+getNumberOfKeys());
24:    }
25:    default void tune() {
26:       play();
27:       printPianoInfo();
28:    } }
```

A. Zero
B. One
C. Two
D. Three
E. Four
F. None of the above

8. How many lines does the following code output?

```
import java.util.*;
public class Exams {
   public static void main(String[] args) {
      List<String> exams = List.of("1Z0-817", "1Z0-819");
      for (var e : exams)
         for (int i=exams.size(); i>0 ; i-=2)
            System.out.print(e+" "+exams.get(i));
            System.out.println();
   }
}
```

A. One.

B. Two.

C. Four.

D. The code does not compile.

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

F. The code compiles but enters an infinite loop at runtime.

9. How many lines of the main method fail to compile?

```
10: public class Transport {
11:    static interface Vehicle {}
12:    static class Bus implements Vehicle {}
13:
14:    public static void main(String[] args) {
15:       Bus bus = new Bus();
16:
17:       System.out.println(null instanceof Bus);
18:       System.out.println(bus instanceof Vehicle);
19:       System.out.println(bus instanceof Bus);
20:       System.out.println(bus instanceof ArrayList);
21:       System.out.println(bus instanceof Collection);
22: } }
```

A. None

B. One

C. Two

D. Three

E. Four

F. Five

10. 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());
```

A. One line needs to be changed for this code to compile.

B. Two lines need to be changed for this code to compile.

C. Three lines need to be changed for this code to compile.

D. If the code is fixed to compile, none of the cells in the 2D array have a value of `0`.

E. If the code is fixed to compile, half of the cells in the 2D array have a value of `0`.

F. If the code is fixed to compile, all of the cells in the 2D array have a value of `0`.

11. How many lines need to be changed to make this method compile?

```
1:  public void colors() {
2:     var yellow = "";
3:     yellow = null;
4:
5:     var red = null;
6:
7:     var blue = "";
8:     blue = 1;
9:
10:    var var = "";
11:    var = "";
12:
13:    var pink = 1;
14: }
```

A. Zero

B. One

C. Two

D. Three

E. Four

F. Five

12. What is the output of the following?

```
10: var result = 8;
11: monitor: while (result >= 7) {
12:    result++;
```

```
13:    do {
14:       result -= 2;
15:       continue monitor;
16:    } while (result > 5);
17: }
18: System.out.println(result);
```

A. 5

B. 6

C. 7

D. The code does not compile.

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

F. The code compiles but enters an infinite loop at runtime.

13. Which of the following lambda expressions can be inserted into both blanks while still allowing the application to compile? (Choose three.)

```
package spooky;
import java.util.function.*;
abstract class Phantom {
   public void bustLater(DoubleConsumer buster, double value) {
      buster.accept(value);
   }
}
public class Ghost extends Phantom {
   public void bustNow(Consumer<Double> buster, double value) {
      buster.accept(value);
   }
   void call() {
      var value = 10.0;
      bustNow(__________________, value);
      bustLater(__________________, value);
   }
}
```

A. `System.out::print`

B. `a -> {System.out.println(a.intValue());}`

C. `g -> {System.out.println();}`

D. `u -> System.out.println((long)u)`

E. `v -> System.out.print(v)`

F. `w -> System.out::println`

14. Given the following class structure, what is the proper way to create an instance of `Spinner` inside the `bake()` method? (Choose three.)

```
public class Kitchen {
   class Mixer {
      class Spinner {}
   }
   public void bake() {
      // INSERT CODE HERE
   }
}
```

A. `var a = new Kitchen().new Mixer().new Spinner();`

B. `Mixer.Spinner b = Mixer.new Spinner();`

C. `var c = new Spinner();`

D. `var d = new Mixer().new Spinner();`

E. `Kitchen.Mixer.Spinner e = new Kitchen().new Mixer().new Spinner();`

F. `Spinner f = new Kitchen().new Mixer().new Spinner();`

15. Which line of code belongs in a service locator?

A. `ServiceLoader<Mouse> sl = ServiceLoader.load(Mouse.class);`

B. `ServiceLoader<Mouse> sl = ServiceLoader.loader(Mouse.class);`

C. `ServiceLoader<Mouse> sl = ServiceLoader.lookup(Mouse.class);`

D. `ServiceLocator<Mouse> sl = ServiceLoader.load(Mouse.class);`

E. `ServiceLocator<Mouse> sl = ServiceLoader.loader(Mouse.class);`

F. `ServiceLocator<Mouse> sl = ServiceLoader.lookup(Mouse.class);`

16. Fill in the blanks: The _____ annotation determines whether annotations are retained in generated Javadoc, while the _____ annotation determines the location an annotation can be applied. (Choose two.)

A. `@Documented` in the first blank

B. `@Javadoc` in the first blank

C. `@Preserve` in the first blank

D. `@Location` in the second blank

E. `@Retention` in the second blank

F. `@Target` in the second blank

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

A. `List<String> a = new ArrayList();`

B. `List<> b = new ArrayList();`

C. `List<String> c = new ArrayList<>();`

D. `List<> d = new ArrayList<>();`

E. `List<String> e = new ArrayList<String>();`

F. `List<> f = new ArrayList<String>();`

18. Which of the following changes, when applied independently, will print the same result as the original implementation? (Choose two.)

```
10: long sum = IntStream.of(4, 6, 8)
11:    .boxed()
12:    .parallel()
13:    .mapToInt(x -> x)
14:    .sum();
15: System.out.print(sum);
```

A. Change the type on line 10 to `double`

B. Change the type on line 10 to `int`

C. Change line 11 to `unboxed()`

D. Remove line 11

E. Remove line 12

F. Remove line 13

19. What does the following code print?

```
// Hare.java
package com.animal;

public class Hare {
   void init() {
      System.out.print("init-");
   }
   protected void race() {
      System.out.print("hare-");
   }
}

// Tortoise.java
package com.animal;

public class Tortoise {
   protected void race(Hare hare) {
      hare.init();    // x1
      hare.race();    // x2
      System.out.print("tortoise-");
    }
```

```
    public static void main(String[] args) {
       var tortoise = new Tortoise();
       var hare = new Hare();
       tortoise.race(hare);
    }
}
```

A. `init-hare-tortoise`

B. `init-hare`

C. The first line with a compiler error is `line x1`.

D. The first line with a compiler error is `line x2`.

E. The code does not compile due to a different line.

F. The code throws an exception.

20. Which statements are true about the `requires` directive? (Choose two.)

A. Changing it to a `requires direct` directive is always allowed.

B. Changing it to a `requires direct` directive is never allowed.

C. Changing it to a `requires direct` directive is sometimes allowed.

D. Including `requires java.base` is allowed, but redundant.

E. Including `requires java.base` is never allowed.

F. Including `requires java.base` is sometimes needed to change the meaning of a file.

21. What is the output of the following?

```
1:  package reader;
2:  import java.util.stream.*;
3:
4:  public class Books {
5:     public static void main(String[] args) {
6:        IntStream pages = IntStream.of(200, 300);
7:        long total = pages.sum();
8:        long count = pages.count();
9:        System.out.println(total + "-" + count);
10:    }
11: }
```

A. `2-2`

B. `200-1`

C. `500-0`

D. `500-2`

E. The code does not compile.

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

22. What is the output of the following?

```
public class InitOrder {
   { System.out.print("1"); }
   static { System.out.print("2"); }

   public InitOrder() {
      System.out.print("3");
   }
   public static void callMe() {
      System.out.print("4");
   }
   public static void main(String[] args) {
      callMe();
      callMe();
      System.out.print("5");
   }
}
```

A. 1223445

B. 2445

C. 22445

D. 223445

E. 2233445

F. None of the above

23. What is the output of the following application? Assume the file system is available and able to be written to and read from.

```
package boat;
import java.io.*;
public class Cruise {
   private int numPassengers = 1;
   private transient String schedule = "NONE";
   { numPassengers = 2; }
   public Cruise() {
      this.numPassengers = 3;
      this.schedule = "Tropical Island";
   }

   public static void main(String... p) throws Exception {
      final String f = "ship.txt";
      try (var o = new ObjectOutputStream(
            new FileOutputStream(f))) {
```

```
      Cruise c = new Cruise();
      c.numPassengers = 4;
      c.schedule = "Casino";
      o.writeObject(c);
   }
   try (var i = new ObjectInputStream(
         new FileInputStream(f))) {
      Cruise c = i.readObject();
      System.out.print(c.numPassengers + "," + c.schedule);
   } } }
```

A. `2,NONE`

B. `3,null`

C. `4,Casino`

D. `4,null`

E. One line would need to be fixed for this code to run without throwing an exception.

F. Two lines would need to be fixed for this code to run without throwing an exception.

24. What does `ServiceLocator.load(ChocolateLab.class)` return?

A. `Collection`

B. `List`

C. `Stream`

D. None of the above

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

A. abstract methods, override, inheritance

B. virtual methods, override, polymorphism

C. concrete methods, overload, inheritance

D. virtual methods, overload, interfaces

E. inheritance, abstract, polymorphism

F. abstract methods, inherit, multiple inheritance

26. What is true about the following?

```
import java.util.*;
public class Yellow {
   public static void main(String[] args) {
      List list = Arrays.asList("Sunny");
      method(list);                                  // c1
   }
   private static void method(Collection<?> x) {   // c2
```

```
      x.forEach(a -> {});                          // c3
  }
}
```

A. The code doesn't compile due to line `c1`.

B. The code doesn't compile due to line `c2`.

C. The code doesn't compile due to line `c3`.

D. The code compiles and runs without output.

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

27. What is the output of the following application?

```
abstract class TShirt {
   abstract int insulate();
   public TShirt() {
      System.out.print("Starting...");
   }
}
public class Wardrobe {
   abstract class Sweater extends TShirt {
      int insulate() {return 5;}
   }
   private void dress() {
      final class Jacket extends Sweater {  // v1
         int insulate() {return 10;}
      };
      final TShirt outfit = new Jacket() {  // v2
         int insulate() {return 20;}
      };
      System.out.println("Insulation:"+outfit.insulate());
   }

   public static void main(String... snow) {
      new Wardrobe().dress();
   }
}
```

A. `Starting...Insulation:20`

B. `Starting...Insulation:40`

C. The code does not compile because of line `v1`.

D. The code does not compile because of line `v2`.

E. The code does not compile for a different reason.

28. Suppose we have a `peacocks` table with two columns: `name` and `rating`. What does the following code output if the table is empty?

```
var url = "jdbc:derby:birds";
var sql = "SELECT name FROM peacocks WHERE name = ?";
try (var conn = DriverManager.getConnection(url);
   var stmt = conn.prepareStatement(sql)) {        // s1

   stmt.setString(1, "Feathers");

   try (var rs = stmt.executeQuery()) {           // s2
      while (rs.hasNext()) {
         System.out.println(rs.next());
      }
   }
}
```

A. `false`

B. `true`

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.

29. 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");
   }
}
```

A. `145`

B. `1045`

C. `24`, followed by a stack trace

D. `245`

E. The code does not compile because of line `p1`.

F. The code does not compile because of line `p2`.

30. What is the output of this code?

```
10: var m = new TreeMap<Integer, Integer>();
11: m.put(1, 4);
12: m.put(2, 8);
13:
14: m.putIfAbsent(2, 10);
15: m.putIfAbsent(3, 9);
16:
17: m.replaceAll((k, v) -> k + 1);
18:
19: m.entrySet().stream()
20:    .sorted(Comparator.comparing(Entry::getKey))
21:    .limit(1)
22:    .map(Entry::getValue)
23:    .forEach(System.out::println);
```

A. `1`

B. `2`

C. `3`

D. `4`

E. The code does not compile.

F. The code compiles but prints something else.

31. Which can fill in the blank so this code outputs `true`?

```
import java.util.function.*;
import java.util.stream.*;

public class HideAndSeek {
   public static void main(String[] args) {
```

```
        var hide = Stream.of(true, false, true);
        Predicate<Boolean> pred = b -> b;
        var found = hide.filter(pred)._______(pred);
        System.out.println(found);
    }
}
```

A. Only `anyMatch`

B. Only `allMatch`

C. Both `anyMatch` and `allMatch`

D. Only `noneMatch`

E. The code does not compile with any of these options.

32. _____ modules are on the classpath, while _____ modules never contain a `module-info` file.

A. Automatic, named

B. Automatic, unnamed

C. Named, automatic

D. Named, unnamed

E. Unnamed, automatic

F. Unnamed, named

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

```
1:  import java.io.*;
2:  class StungException extends Exception {}
3:  class Suit implements Closeable {
4:     public void close() throws IOException {}
5:  }
6:  public class BeeCatcher {
7:     public static void main(String[] b) throws IOException {
8:        var s = new Suit();
9:        var t = new Suit();
10:       try (s; t) {
11:          throw new StungException();
12:       } catch (StungException | Exception e) {
13:          s = null;
14:       } finally {
15:       }
16:    }
17: }
```

A. One
B. Two
C. Three
D. Four
E. None. The code compiles as is.

34. What is the output of the Light program?

```
package physics;
class Wave {
   public int size = 7;
}
public class Light extends Wave {
   public int size = 5;
   public static void main(String... emc2) {
      Light v1 = new Light();
      var v2 = new Light();
      Wave v3 = new Light();
      System.out.println(v1.size +","+ v2.size +","+ v3.size);
   }
}
```

A. `5,5,5`
B. `5,5,7`
C. `5,7,7`
D. `7,7,7`
E. The code does not compile.
F. None of the above.

35. How many of the following could be valid JDBC URL formats for an imaginary driver named `magic` and a database named `box`?

```
String first = "jdbc;box;magic";
String second = "jdbc;magic;@127.0.0.1:1234";
String third = "jdbc;magic;127.0.0.1:1234/box";
```

A. Only `first`
B. Only `second`
C. Only `third`
D. `first` and `second`
E. `first` and `third`
F. None of these

36. Which of the following statements about interface methods are correct? (Choose three.)

A. A `private static` interface method can call `default` methods.

B. A `public static` interface method can call `abstract` methods.

C. A `private static` interface method can call `static` methods.

D. A `default` interface method can call `private static` methods.

E. A `default` interface method can call `abstract` methods.

F. A `public static` interface method can call `default` methods.

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

```
public static void main(String[] args) {
   String state1 = new String("ice");
   String state2 = new String("water");
   String state3 = new String("mist");

   state1 = state2;
   state2 = state3;
   state3 = state1;
}
```

A. No objects are eligible for garbage collection.

B. One object is eligible for garbage collection.

C. Two objects are eligible for garbage collection.

D. No objects are guaranteed to be garbage collected.

E. One object is guaranteed to be garbage collected.

F. Two objects are guaranteed to be garbage collected.

38. Which of the following can fill in the blank to output `sea lion, bald eagle`?

```
String names = Stream.of(
   "bald eagle", "pronghorn", "puma", "sea lion")

   ________________

   .collect(Collectors.joining(", "));
System.out.println(names);
```

A.

```
.filter(s -> s.contains(" "))
.collect(Collectors.toSet())
.stream()
.entrySet()
.stream()
```

```
.filter(e -> e.getKey())
.map(Entry::getValue)
.flatMap(List::stream)
.sorted(Comparator.reverseOrder())
```

B.

```
.filter(s -> s.contains(" "))
.collect(Collectors.toUnmodifiableSet())
.stream()
.entrySet()
.stream()
.filter(e -> e.getKey())
.map(Entry::getValue)
.flatMap(List::stream)
.sorted(Comparator.reverseOrder())
```

C.

```
.collect(Collectors.toUnmodifiableSet())
.stream()
.collect(Collectors.groupingBy(s -> s.contains(" ")))
.entrySet()
.stream()
.filter(e -> e.getKey())
.map(Entry::getValue)
.map(List::stream)
.sorted(Comparator.reverseOrder())
```

D.

```
.collect(Collectors.toSet())
.stream()
.collect(Collectors.groupingBy(s -> s.contains(" ")))
.entrySet()
.stream()
.filter(e -> e.getKey())
.map(Entry::getValue)
.flatMap(List::stream)
.sorted(Comparator.reverseOrder())
```

E.

```
.filter(s -> s.contains(" "))
.collect(Collectors.toUnmodifiableSet())
.stream()
.collect(Collectors.groupingBy(s -> s.contains(" ")))
```

```
.entrySet()
.stream()
.filter(e -> e.getKey())
.map(Entry::getValue)
.map(List::stream)
.sorted(Comparator.reverseOrder())
```

F.

```
.collect(Collectors.toUnmodifiableSet())
.stream()
.collect(Collectors.groupingBy(s -> s.contains(" ")))
.entrySet()
.stream()
.filter(e -> e.getKey())
.map(Entry::getValue)
.map(List::stream)
.sorted(Comparator.reverseOrder())
```

39. What is the minimum number of lines that need to be removed to make this code compile and be able to implemented as a lambda expression?

```
@FunctionalInterface
public interface Play {
   public static void baseball() {}
   private static void soccer() {}
   default void play() {}
   void fun();
   void game();
   void toy();
}
```

A. 1

B. 2

C. 3

D. 4

E. The code compiles as is.

40. Which message does the following application print?

```
package ranch;
public class Cowboy {
   private int space = 5;
   private double ship = space < 2 ? 3L : 10.0f;  // g1
   public void printMessage() {
```

```
      if(ship>1) {
         System.out.print("Goodbye!");
      } if(ship<10 && space>=2)                    // g2
         System.out.print("Hello!");
      else System.out.print("See you again!");
   }
   public static final void main(String... stars) {
      new Cowboy().printMessage();
   }
}
```

A. `Hello!`
B. `Goodbye!`
C. `See you again!`
D. It does not compile because of line `g1`.
E. It does not compile because of line `g2`.
F. None of the above.

41. Which are included in the Java Platform Module System? (Choose three.)

A. A format for module JARs
B. A list of all possible modules for Java
C. A new file format called `jdeps`
D. Additional command-line options for Java tools
E. Decommissioning of the `jar` format
F. Partitioning of the JDK into modules

42. 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);
   }
}
```

A. `public Big(boolean stillIn)`

B. `public Trouble()`

C. `public Trouble(int deep)`

D. `public Trouble(String now, int... deep)`

E. `public Trouble(long deep)`

F. `public Trouble(double test)`

43. Fill in the blanks: The name of the `abstract` method in the `Function` interface is _____, while the name of the `abstract` method in the `Consumer` interface is _____.

A. `accept()`, `apply()`

B. `accept()`, `get()`

C. `apply()`, `accept()`

D. `apply()`, `apply()`

E. `apply()`, `test()`

44. How many lines fail to compile?

```
class Roller<E extends Wheel> {
   public void roll(E e) { }
}
class Wheel { }
class CartWheel extends Wheel { }

public class RollingContest {
   Roller<CartWheel> wheel1 = new Roller<CartWheel>();
   Roller<Wheel> wheel2 = new Roller<CartWheel>();
   Roller<? extends Wheel> wheel3 = new Roller<CartWheel>();
   Roller<? extends Wheel> wheel4 = new Roller<Wheel>();
   Roller<? super Wheel> wheel5 = new Roller<CartWheel>();
   Roller<? super Wheel> wheel6 = new Roller<Wheel>();
}
```

A. One

B. Two

C. Three
D. Four
E. Five
F. Six

45. What is the output of the following program?

```
var bed = List.of((short)2,(short)5);
var pillow = bed.parallelStream().reduce(0,
   (a,b) -> b.doubleValue() + a.doubleValue(),
   (c,d) -> d.doubleValue() + c.doubleValue());
System.out.println(pillow);
```

A. `0`
B. `0.0`
C. `7`
D. `7.0`
E. The code does not compile
F. None of the above

46. Given the following three property files, what does the following method output?

toothbrush.properties

```
color=purple
type=generic
```

toothbrush_es.properties

```
color=morado
type=lujoso
```

toothbrush_fr.properties

```
color=violette
```

```
void brush() {
   Locale.setDefault(new Locale.Builder()
      .setLanguage("es")
      .setRegion("MX").build());
   var rb = ResourceBundle.getBundle("toothbrush",
      new Locale("fr"));
   var a = rb.getString("color");
   var b = rb.getString("type");
   System.out.print(a + " " + b);
}
```

A. `morado null`

B. `violette generic`

C. `morado lujoso`

D. `violette null`

E. The code does not compile.

F. An exception is thrown at runtime.

47. Which of the following are valid functional interfaces in the `java.util.function` package? (Choose three.)

A. `BooleanSupplier`

B. `CharSupplier`

C. `DoubleUnaryOperator`

D. `ObjectIntConsumer`

E. `ToLongBiFunction`

F. `TriPredicate`

48. What change, if any, should be made to the following method to improve security?

```
10: public List<String> accessNetworkList(String fileName) {
11:    return AccessController.doPrivileged(
12:          new PrivilegedAction<List<String>>() {
13:       public List<String> run() {
14:          try {
15:             return Collections.unmodifiableList(
16:                Files.readAllLines(Path.of(fileName)));
17:          } catch (IOException e) {
18:             throw new SecurityException("No access");
19:          } } });
20: }
```

A. On line 10, the method should be marked `private`.

B. On line 15, an `ArrayList` instance should be returned instead of an unmodifiable list.

C. Prior to line 16, the `fileName` should be validated against a list of constants.

D. The exception on line 18 should be removed and an empty `List` should be returned.

E. None of the above, as the code is safe as is.

49. What is the result of executing the `Clownfish` program?

```
package ocean;
class BubbleException extends Exception {}
abstract class Fish {
   Fish getFish() {
```

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

A. The code compiles and prints `swim!`

B. The code compiles and prints `fish!`

C. The code compiles and prints a stack trace.

D. One line of the program does not compile.

E. Two lines of the program do not compile.

F. None of the above.

50. What statements about the following method are correct? (Choose three.)

```
public String findWaffles(String connectionStr, String search)
      throws SQLException {
   var query = "SELECT * FROM meal WHERE type = '"+search+"'";
   var con = DriverManager.getConnection(connectionStr);
   try (con;
         var ps = con.prepareStatement(query);
         var rs = ps.executeQuery()) {
      return rs.getString("name");
   } }
```

A. It protects against a denial of service attack.

B. It does not protect against denial of service attacks.

C. It protects against SQL injection because it uses a `PreparedStatement`.

D. It does not protect against SQL injection.

E. Assuming the database and related table exist and are available, this mode is expected to run without any exceptions being thrown.

F. This method will always produce an exception at runtime.