OCA / OCP Java SE 8 Programmer Practice Tests

PREV
Chapter 22 Localization

NEXT
Appendix Answers to Review Questions

## Chapter 23
## OCP Practice Exam

This chapter contains 85 questions and is designed to simulate a real OCP 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. Suppose the `ResultSet` is scrollable and contains 10 rows with values numbered from 1 to 10 in ascending order. What is true about the following? (Choose two.)

   ```
   rs.beforeFirst();
   System.out.println(rs.relative(5));
   System.out.println(rs.getInt(1));
   System.out.println(rs.relative(-10));
   System.out.println(rs.getInt(1));
   System.out.println(rs.relative(5));
   System.out.print(rs.getInt(1));
   ```

   1. It outputs `true` once.
   2. It outputs `true` twice.
   3. It outputs `true` three times.
   4. It completes without throwing an exception.
   5. It throws an exception at runtime.

2. What is the result of the following?

   ```
   public class PiDay {
      public static void main(String[] args) {
         LocalDateTime pi = LocalDateTime.of(2017, 3, 14, 1, 59);
         DateTimeFormatter formatter = DateTimeFormatter
            .ofPattern("m.ddhhMM");
         System.out.println(formatter.format(pi));
      }
   }
   ```

   1. `3.011459`
   2. `3.140159`

5. The code does not compile.
6. The code compiles but throws an exception at runtime.

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

1. `() -> {}`
2. `(Double adder) -> {int y; System.out.print(adder); return adder;}`
3. `(Long w) -> {Long w=5; return 5;}`
4. `(int count, vote) -> count*vote`
5. `dog -> dog`
6. `name -> {name.toUpperCase()}`

4. 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.p
   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);
   }
}
```

1. It prints two lines.
2. It prints three lines.
3. One line of code does not compile.
4. Two lines of code do not compile.
5. Three lines of code do not compile.
6. The code compiles but prints an exception at runtime.

5. What is the output of the following?

```
public class Compete {
   public static void main(String[] args) {
      Stream<Integer> is = Stream.of(8, 6, 9);
      Comparator<Integer> c = (a, b) -> b - a;  // r1
      is.sorted(c).forEach(System.out::print);  // r2
  }
}
```

1. `689`
2. `986`
3. The code does not compile because of line `r1`.
4. The code does not compile because of line `r2`.
5. The code does not compile due to another line.
6. The code compiles but throws an exception at runtime.

6. What is the output of the following application?

```
package tax;

public class Accountant {
   class AddingException extends Exception {};
   class DividingException extends Exception {};
   class UnexpectedException extends RuntimeException {};
   public void doTaxes() throws Throwable {
      try {
         throw new IllegalStateException();
      } catch (AddingException | DividingException e) {  // p1
```

```
         System.out.println("Math Problem");
      } catch (UnexpectedException | Exception f) {  // p2
         System.out.println("Unknown Error");
         throw f;
      }
   }
   public static void main(String[] numbers) throws Throwable {
      try {
         new Accountant().doTaxes();
      } finally {
         System.out.println("All done!");
      }
   }
}
```

1. `Math Problem`
2. `Unknown Problem`
3. `Unknown Problem` followed by `All done!`
4. The code does not compile solely due to line `p1`.
5. The code does not compile solely due to line `p2`.
6. The code does not compile due to lines `p1` and `p2`.

7. What is the output of the following application?

```
package drawing;
interface HasHue {String getHue();}
enum COLORS implements HasHue {
   red {
      public String getHue() {return "FF0000";}
   }, green {
      public String getHue() {return "00FF00";}
   }, blue {
      public String getHue() {return "0000FF";}
   }
}
class Book {
   static void main(String[] pencils) {}
}
final public class ColoringBook extends Book {
   final void paint(COLORS c) {
      System.out.print("Painting: "+c.getHue());
   }
   final public static void main(String[] crayons) {
      new ColoringBook().paint(COLORS.green);
   }
}
```

1. `Painting: 00FF00`
2. One line of code does not compile.
3. Two lines of code do not compile.
4. Three lines of code do not compile.
5. The code compiles but prints an exception at runtime.
6. None of the above

8. How many of the following can fill in the blank to make this code compile?

```
public boolean isItMyBirthday(LocalDateTime dateTime) {
   ________________________________________
   return now.getMonth() == dateTime.getMonth()
      && now.getDayOfMonth() == dateTime.getDayOfMonth();
}
```

1. `LocalDate now = LocalDate.now();`
2. `LocalDate now = new LocalDate();`
3. `LocalDateTime now = LocalDateTime.now();`
4. `LocalDateTime now = new LocalDateTime();`
5. `ZonedDate now = ZonedDate.now();`
6. `ZonedDate now = new ZonedDate();`

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

9. Which two can independently fill in the blank to output `No dessert today`? (Choose two.)

```
import java.util.*;
public class Dessert {
  public static void main(String[] yum) {
    eatDessert(Optional.empty());
  }
  private static void eatDessert(Optional<String> opt) {
     System.out.println(opt.__________);
  }
}
```

1. `get("No dessert today")`
2. `get(() -> "No dessert today")`
3. `orElse("No dessert today")`
4. `orElse(() -> "No dessert today")`
5. `orElseGet("No dessert today")`
6. `orElseGet(() -> "No dessert today")`

10. 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");
   }
}
```

1. `1223445`
2. `2445`
3. `22445`
4. `223445`
5. `2233445`
6. None of the above

11. Which of the following cannot be instantiated directly by the caller using the constructor?

1. `Locale`
2. `Properties`
3. `ResourceBundle`

1. I
2. II
3. III
4. I, II
5. I, III
6. II, III

12. What design pattern or principle is used in this class?

```
public class Daffodil {
   int height;
   public Daffodil(int height) {
      this.height = height;
   }
   public int getHeight() {
      return height;
   }
}
```

1. Encapsulation
2. Immutability
3. Singleton

4. Both A and B
5. None of the above

13. 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... passengers) throws Exception {
      try (ObjectOutputStream o = new ObjectOutputStream(
            new FileOutputStream("ship.txt"))) {
         Cruise c = new Cruise();
         c.numPassengers = 4;
         c.schedule = "Casino";
         o.writeObject(c);
      }
      try (ObjectInputStream i = new ObjectInputStream(
            new FileInputStream("ship.txt"))) {
         Cruise c = i.readObject();
         System.out.print(c.numPassengers+","+c.schedule);
      }
   }
}
```

1. `2,NONE`
2. `3,null`
3. `4,Casino`
4. `4,null`
5. The class does not compile.
6. The class compiles but throws an exception at runtime.

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

1. `Driver`
2. `DriverManager`
3. `Query`
4. `ResultSet`

1. I, III
2. I, IV
3. II, III
4. II, IV
5. I, II, III
6. I, II, IV

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

1. `v -> {System.out.print("Hello!"); return 2%1;}`
2. `(Integer w) -> w.intValue()`
3. `(int j) -> (int)30L`
4. `(int q) -> q/3.1`
5. `(long x) -> (int)x`
6. `z -> z`

16. Which of the following statements about `InputStream` and `Reader` are correct? (Choose two.)

1. One contains a `read()` method that returns a `byte` value, while the other contains a `read()` method that returns an `int` value.
2. Only one of them contains a `flush()` method.
3. Only one of them contains a `skip()` method.

4. They are both `abstract` classes.
5. They are both interfaces.
6. They can both be used to read character data.

17. Fill in the blank so this code prints `-1`.

```
LocalDate xmas = LocalDate.of(2017,  12, 25);
LocalDate blackFriday = LocalDate.of(2017, 11, 24);
long monthsLeft = ChronoUnit.MONTHS.____________________;
System.out.println(monthsLeft);
```

1. `between(blackFriday, xmas)`
2. `between(xmas, blackFriday)`
3. `minus(blackFriday, xmas)`
4. `minus(xmas, blackFriday)`
5. `plus(blackFriday, xmas)`
6. `plus(xmas, blackFriday)`

18. Which statements about the following class are true?

```
package secure;
import java.io.*;
public class Login {
   public void clearPassword(char[] password) {
      for(int i=0; i<password.length; i++) {
         password[i] = 0;
      }
   }
   public String getPassword() {
      Console c = System.console();
      final char[] pass = c.readPassword("Enter your password: ");
      StringBuilder sb = new StringBuilder();
      for(char p : pass) {
         sb.append(p);
      }
      clearPassword(pass);
      return sb.toString();
   }
   public static void main(String[] webLogin) {
      String pass = new Login().getPassword();
   }
}
```

1. The class compiles.
2. The design protects the password by clearing it from memory after it is entered.
3. The class may throw an exception at runtime.

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

19. Which of the following can fill in the blank to print out the numbers `161`, `183`, and `201` in any order?

```
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) {
      Stream<Runner> runners = Stream.of(new Runner(183),
         new Runner(161), new Runner(201));
      OptionalInt opt = runners.____________________;
   }
}
```

1. `map(Runner::getNumberMinutes)`
1. `.peek(System.out::println).max()`

2. `mapToInt(Runner::getNumberMinutes)`
1. `.peek(System.out::println).max()`
3. `peek(System.out::println)`
1. `.mapToInt(Runner::getNumberMinutes).max()`
4. `peek(System.out::println)`
1. `.mapToInt(Runner::getNumberMinutes).max()`
5. None of the above

20. What is the output of the following application?

```
package fruit;
enum Season {
   SPRING(1), SUMMER(2), FALL(3), WINTER(4)
   public Season(int orderId) {}
}
public class PickApples {
   public static void main(String... orchard) {
      final Season s = Season.FALL;
      switch(s) {
         case 3:
            System.out.println("Time to pick!");
         default:
            System.out.println("Not yet!");
      }
   }
}
```

1. `Time to pick!`
2. `Time to pick!` followed by `Not yet!`
3. One line of code does not compile.
4. Two lines of code do not compile.
5. Three lines of code do not compile.
6. The code compiles but prints an exception at runtime.

21. Which of the following expressions, inserted simultaneously into both blanks, allow 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() {
      double value = 10;
      bustNow(__________,value);
      bustLater(_________,value);
   }
}
```

1. `System.out::print`
2. `a -> {System.out.println(a.intValue());}`
3. `g -> {System.out.println();}`
4. `u -> System.out.println((long)u)`
5. `v -> System.out.print(v)`
6. `w -> System.out::println`

22. What is the output of the following?

```
package counter;
import java.util.*;

public class CountResource extends ListResourceBundle {
   private int count = 0;

   @Override
   protected Object[][] getContents() {
      return new Object[][] { { "count", ++count } };
   }
   public static void main(String[] args) {
      ResourceBundle rb = ResourceBundle.getBundle("counter.CountRe
      System.out.println(rb.getString("count") + " " + rb.getString
   }
}
```

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

23. In most of the United States, daylight savings time ends on November 5, 2017 at 02:00 a.m., and we repeat that hour. What is the output of the following?

```
import java.time.*;

public class FallBack {

   public static void main(String[] args) {
      LocalDate localDate = LocalDate.of(2017, 10, 5);
      LocalTime localTime = LocalTime.of(1, 0);
      ZoneId zone = ZoneId.of("America/New_York");
      ZonedDateTime z = ZonedDateTime.of(localDate, localTime, zone

      for (int i = 0; i < 6; i++)
         z = z.plusHours(1);

      System.out.println(z.getHour());
   }
}
```

1. `5`
2. `6`
3. `7`
4. The code does not compile.
5. The code compiles, but throws an exception at runtime.

24. Assume the `/environment` directory exists and contains a file with a symbolic link to the `/environment` directory. In addition, assume all files within the directory are fully accessible. What is the result of executing the following program?

```
import java.nio.file.*;
import java.nio.file.attribute.*;
public class SearchEnvironment {
   public static void accessFile(Path p, long timeEpoch) {
      try {
         Files.readAttributes(p, BasicFileAttributes.class)
            .setTimes(null, null, FileTime.fromMillis(timeEpoch));
      } catch (Throwable e) {
      } finally {}
   }
   public static final void main(String[] unused) throws Exception
      Path w = Paths.get("/environment");
      Files.walk(w)
         .forEach(q -> accessFile(q,System.currentTimeMillis()));
   }
}
```

1. The code does not compile.
2. The program exits after successfully updating the creation time of files within the directory.
3. The program exits after successfully updating the last accessed time of files within the directory.
4. The program compiles but throws an exception at runtime.
5. The program enters an infinite loop and hangs at runtime.

25. Which command causes the following class to throw an `AssertionError` at runtime?

```
public class Watch {
   private static final short DEFAULT_HOUR = 12;
   private Watch() {
      super();
   }
   int checkHour() {
      assert DEFAULT_HOUR > 12;
      return DEFAULT_HOUR;
   }
   public static void main(String... ticks) {
      new Watch().checkHour();
   }
}
```

1. `java -da Watch`
2. `java -ea:Watch -da Watch`
3. `java -ea -da:Watch Watch`
4. `java -enableassert Watch`
5. None of the above since the `Watch` class does not compile.

26. What is the output of the following application?

```
package rope;
import java.util.concurrent.*;
import java.util.stream.IntStream;
public class Jump {
   private static void await(CyclicBarrier b) {
      try {
         b.await();
      } catch (InterruptedException | BrokenBarrierException e) {
         e.printStackTrace();
      }
   }
   public static void main(String[] chalk) {
      ExecutorService s = Executors.newFixedThreadPool(4);
      final CyclicBarrier b = new CyclicBarrier(4,
         () -> System.out.print("Jump!"));
      IntStream
         .iterate(1, q -> 2)
         .limit(10)
         .forEach(q -> s.execute(() ->await(b)));
      s.shutdown();
   }
}
```

1. `Jump!` is printed and exits.
2. `Jump!` is printed twice and exits.
3. The code does not compile.
4. The output cannot be determined ahead of time.
5. The program hangs indefinitely at runtime because the `IntStream` is not parallel.
6. None of the above

27. What is the output of the following code snippet? Assume the two directories referenced both exist and are symbolic links to the same location within the file system.

```
if(Files.isSameFile("/salad/carrot", "/fruit/apple"))
   System.out.println("Same!");
else System.out.println("Different!");
```

1. `Same!`
2. `Different!`
3. The code does not compile.
4. The code compiles but throws an exception at runtime.
5. None of the above

28. Which can fill in the blank `JavaProgrammerCert` class to compile and logically complete the code? (Choose two.)

```
class JavaProgrammerCert extends Exam {
   private Exam oca;
   private Exam ocp;
   // assume getters and setters are here
   ______________________________________
}
public class Exam {
   boolean pass;
   protected boolean passed() {
     return pass;
   }
}
```

1. `boolean passed() { return oca.pass && ocp.pass; }`

2. `boolean passed() { return oca.passed() && ocp.passed(); }`
3. `boolean passed() { return super.passed(); }`
4. `public boolean passed() { return oca.passed() && ocp.passed(); }`
5. `public boolean passed() { return oca.pass && ocp.pass; }`
6. `public boolean passed() { return super.passed(); }`

29. Which statements about overriding a method are correct? (Choose two.)

1. An overridden method must not throw a narrower version of any checked exception thrown by the inherited method.
2. An overridden method must use a return type that is covariant with the inherited method.
3. An overridden method must use a different set of input arguments as the inherited method.
4. Overridden methods must use an access modifier at least as broad as their inherited method.
5. It is possible to override `private` and `static` methods.
6. The `@Override` annotation is required anytime a method is overridden.

30. Which of the following can be independently inserted into the blank so the code can run without error for at least one SQL query?

```
private static void choices(Connection conn, String sql)       thro
   try (Statement stmt = conn.createStatement();
       ResultSet rs = stmt.executeQuery(sql)) {
       ___________________

   }
}
```

1. `System.out.println(rs.getInt(1));`
2. `rs.next(); System.out.println(rs.getInt(1));`
3. `if (rs.next()) System.out.println(rs.getInt(1));`

1. II
2. III
3. I and III
4. II and III
5. I, II, and III
6. None of the above

31. Starting with `DoubleBinaryOperator` and going downward, fill in the values for the table.

| **Functional Interface** | **# Parameters** |
|---|---|
| `DoubleBinaryOperator` | |
| `LongToIntFunction` | |
| `ToLongBiFunction` | |
| `IntSupplier` | |
| `ObjLongConsumer` | |

1. `1, 0, 0, 0, 2`
2. `1, 2, 1, 0, 1`
3. `2, 1, 0, 1, 2`
4. `2, 1, 1, 0, 1`

5. `2, 1, 2, 0, 2`
6. `3, 0, 2, 1, 1`

32. What is a possible output of the following?

```
LocalDate trainDay = LocalDate.of(2017, 5, 13);
LocalTime time = LocalTime.of(10, 0);
ZoneId zone = ZoneId.of("America/Los_Angeles");
ZonedDateTime zdt = ZonedDateTime.of(trainDay, time, zone);
Instant instant = zdt.toInstant();
instant = instant.plus(1, ChronoUnit.YEARS);
System.out.println(instant);
```

1. `2017-05-13T10:00-07:00[America/Los_Angeles]`
2. `2017-05-13T17:00:00Z`
3. `2018-05-13T10:00-07:00[America/Los_Angeles]`
4. `2018-05-13T17:00:00Z`
5. The code does not compile.
6. The code compiles but throws an exception at runtime.

33. How many lines does the following output?

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

class Blankie {
   String color;
   boolean isPink() {
      return "pink".equals(color);
   }
}
public class PreSchool {
   public static void main(String[] args) {
     Blankie b1 = new Blankie();
     Blankie b2 = new Blankie();
     b1.color = "pink";
     Stream.of(b1, b2).filter(Blankie::isPink).forEach(System.out::
   }
}
```

1. None
2. One
3. Two
4. The code does not compile.
5. The code compiles but throws an exception at runtime.

34. Which are the minimum changes needed to make this class immutable?

```
public class Tree {
   String species;
   public Tree(String species) {
      this.species = species;
   }
   public String getSpecies() {
     return species;
   }
   private final void setSpecies(String newSpecies) {
      species = newSpecies;
  }
}
```

1. Make `species private` and `final`.
2. Make the getter method final.
3. Remove the setter method.

1. None. It is already immutable.
2. I
3. I and II
4. I and III
5. II and III
6. I, II, and III

35. Let’s say you needed to write a large `List` of `Student` objects to a data file. Which three concrete classes, chained together, would best accomplish this? (Choose three.)

1. `BufferedFileOutputStream`
2. `BufferedOutputStream`
3. `FileOutputStream`
4. `FileWriter`
5. `ObjectOutputStream`
6. `OutputStream`

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

```
ArrayDeque<Integer> d = new ArrayDeque<>();
d.offer(18);
// INSERT CODE HERE
```

1. `d.peek(); d.peek();`
2. `d.poll(); d.poll();`
3. `d.pop(); d.pop();`
4. `d.remove(); d.remove();`

37. 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.relativize(x));
```

1. `/./song/../note/dance/move.txt`
1. `/dance/move.txt`
2. `/dance/move.txt`
1. `/dance/move.txt/./song/../note`
3. `/dance/move.txt`
1. `/dance/move.txt/note`
4. `/note/dance/move.txt`
1. `../dance/move.txt/song`
5. The code does not compile.
6. The code compiles but an exception is thrown at runtime.

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

```
package field;
import java.io.*;
class StungException extends Exception {}
class Suit implements Closeable {
   public void close() throws IOException {}
}
public class BeeCatcher {
   public static void main(String... bees) {
      try (Suit s = new Suit(), Suit t = new Suit()) {
         throw new StungException();
      } catch (Exception e) {
      } catch (StungException e) {
      } finally {
      }
   }
}
```

1. One
2. Two
3. Three
4. Four
5. None. The code compiles as is.

39. 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) {
      Stream<Integer> s1 = Arrays.asList(1,2,3,4,5).stream();
      Stream<Integer> s2 = Arrays.asList(1,2,3,4,5).stream();
      int val1 = findFast(s1);
      int val2 = findSlow(s2);
      System.out.print(val1+" "+val2);
   }
}
```

1. `1 1`
2. `3 1`
3. The answer cannot be determined until runtime.
4. The code does not compile.
5. The code compiles but throws an exception at runtime.

40. Given this property file used to load the `Properties` object `props` and this code snippet, what is the output?

```
mystery=bag
type=paper

System.out.print(props.getDefaultProperty("mystery", "?"));
System.out.print(" ");
System.out.print(props.getDefaultProperty("more", "?"));
```

1. `bag`
2. `bag null`
3. `bag ?`
4. The code does not compile.
5. The code compiles but throws an exception at runtime.

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

1. Any resource used must implement `Closeable`.
2. 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.
3. If the `try` block and `close()` method both throw an exception, the one thrown by the `try` block is suppressed.
4. Neither a `catch` nor a `finally` block is required.
5. The `close()` method of the resources must throw a checked exception.

42. 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>();
}
```

1. One
2. Two
3. Three
4. Four
5. Five
6. Six

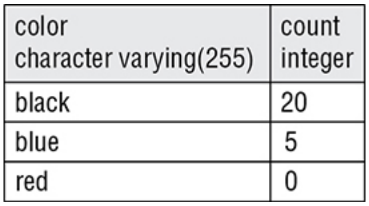
43. Which are the minimum changes needed to properly implement the singleton pattern?

```
1:   public class Bookmark {
2:      private static Bookmark bookmark;
3:      private int pageNumber;
4:      static {
5:        bookmark = new Bookmark();
6:      }
7:      public static Bookmark getInstance() {
8:         return bookmark;
9:      }
10:     public int getPageNumber() {
11:        return pageNumber;
12:     }
13:     public void setPageNumber(int newNumber) {
14:        pageNumber = newNumber;
15:     }
16:  }
```

1. Add a private constructor.
2. Remove the setter method.
3. Remove the `static` block and change line 2 to instantiate `Bookmark`.

1. None. It is already a singleton.
2. I
3. I and II
4. I and III
5. II and III
6. I, II, and III

44. Given an updatable `ResultSet` that contains the following and this code, what does the code snippet output?



| color<br>character varying(255) | count<br>integer |
|---|---|
| black | 20 |
| blue | 5 |
| red | 0 |

```
rs.afterLast();
rs.previous();
rs.updateInt(2, 10);

rs = stmt.executeQuery("select * from pens where color = 'red'");
while (rs.next()) {
   System.out.println(rs.getInt(2));
}
```

1. `0`
2. `10`
3. The code does not compile.
4. The code compiles but throws an exception at runtime.

45. Which statements describe a `java.io` stream class and cannot be applied to a `java.util.stream.Stream` class? (Choose three.)

1. Can be used with try-with-resources statement
2. Includes a class or set of classes used solely for working with character data
3. Requires all data objects to implement `Serializable`
4. Some classes contain a `flush()` method.
5. Some classes contain a method to skip over data.
6. Some classes contain a method to sort the data.

46. 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?

```
package bookworm;
import java.io.*;
import java.nio.file.*;
public class ReadEverything {
   public void readFile(Path p) {
      try {
         Files.readAllLines(p)
         .parallel()
         .forEach(System.out::println);
      } catch (Exception e) {}
   }
   public void read(Path directory) throws Exception {
      Files.walk(directory)
         .filter(p -> File.isRegularFile(p))
         .forEach(x -> readFile(x));
   }
   public static void main(String... books) throws IOException
      Path p = Path.get("collection");
      new ReadEverything().read(p);
   }
}
```

1. None. Bill’s implementation is correct.
2. One
3. Two
4. Three
5. Four
6. Five

47. Assuming the following class is concurrently accessed by numerous threads, which statement about the `CountSheep` class is correct?

```
package fence;
import java.util.concurrent.atomic.*;
public class CountSheep {
   private static AtomicInteger counter = new AtomicInteger();
   private Object lock = new Object();
   public synchronized int increment1() {
      return counter.incrementAndGet();
   }
   public static synchronized int increment2() {
      return counter.getAndIncrement();
   }
   public int increment3() {
      synchronized(lock) {
         return counter.getAndIncrement();
      }
   }
}
```

1. The class is thread-safe only if `increment1()` is removed.
2. The class is thread-safe only if `increment2()` is removed.
3. The class is thread-safe only if `increment3()` is removed.
4. The class is already thread-safe.
5. The class does not compile.
6. The class compiles but may throw an exception at runtime.

48. Which of the following are not required parameters for the NIO.2 `Files.find()` method? (Choose two.)

1. `BiPredicate`
2. `FileVisitOption...`
3. `int`
4. `long`
5. `Path`

49. Which statements are correct? (Choose two.)

1. A `Comparable` implementation is often implemented by a lambda.
2. A `Comparable` object has a `compare()` method.
3. The `compare()` and `compareTo()` methods have the same contract for the return value.
4. There can be multiple `Comparator` implementations for the same class.

5. Two objects that return `true` for `equals()` will always return `0` when passed to `compareTo()`.

50. What is the output of the following code snippet, assuming none of the files referenced exist within the file system?

```
Path t1 = Paths.get("/sky/.././stars.exe");
Path t2 = Paths.get("/stars.exe");
Path t3 = t1.resolve(t2);

boolean b1 = t1.equals(t2);
boolean b2 = t1.normalize().equals(t2);
boolean b3 = Files.isSameFile(t1.normalize(),t2);
boolean b4 = Files.isSameFile(t2,t3);

System.out.print(b1+","+b2+","+b3+","+b4);
```

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

51. Let's say we have a `Reader` instance that will produce the characters with the numeric values `{1,2,3,4,5,6,7}`. Which of the following are possible outcomes of executing the `checkLottoNumbers()` method with this `Reader` instance? (Choose two.)

```
23: public String checkLottoNumbers(Reader r) throws IOException {
24:    r.read();r.skip(1);
25:    r.mark(5);
26:    r.skip(1);
27:    r.reset();
28:    return r.read()+"-"+r.read(new char[5]);
29: }
```

1. An `IOException` on line 25
2. An `IOException` on line 27
3. `'c'-4` is returned.
4. `'d'-3` is returned.
5. `3-4` is returned.
6. `4-3` is returned.

52. 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___________ .

1. `accept()`, `apply()`
2. `accept()`, `get()`
3. `apply()`, `accept()`
4. `apply()`, `apply()`
5. `apply()`, `test()`

53. Assuming the following program is executed with assertions enabled, which is the first line to throw an exception at runtime?

```
1:  package school;
2:  public class Teacher {
3:     public int checkClasswork(int choices) {
4:        assert choices++==10 : 1;
5:        assert true!=false : new StringBuilder("Answer2");
6:        assert(null==null) : new Object();
7:        assert ++choices==11 : "Answer4";
8:        assert 2==3 : "";
9:        return choices;
10:    }
11:    public final static void main(String... students) {
12:       try {
13:          new Teacher().checkClasswork(10);
14:       } catch (Error e) {
15:          System.out.print("Bad idea");
16:          throw e;
17:       }
18:    }
19: }
```

1. Line 4
2. Line 5
3. Line 6
4. Line 7
5. Line 8
6. None of the above since the class does not compile

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

1. `BooleanSupplier`
2. `CharSupplier`
3. `DoubleUnaryOperator`
4. `ObjectIntConsumer`
5. `ToLongBiFunction`
6. `TriPredicate`

55. Which statements about the following class are correct? (Choose two.)

```
package knowledge;
class InformationException extends Exception {}
public class LackOfInformationException extends InformationExceptio
   public LackOfInformationException() {  // t1
      super("");
   }
   public LackOfInformationException(String s) {  // t2
      this(new Exception(s));
   }
   public LackOfInformationException(Exception c) {  // t3
      super();
   }
   @Override public String getMessage() {
      return "lackOf";
   }
}
```

1. `LackOfInformationException` compiles without issue.
2. The constructor declared at line `t1` does not compile.
3. The constructor declared at line `t2` does not compile.
4. The constructor declared at line `t3` does not compile.
5. The `getMessage()`method does not compile because of the `@Override` annotation.
6. `LackOfInformationException` is a checked exception.

56. How many changes do you need to make in order for this code to compile?

```
public class Ready {
   private static double getNumber() {
      return .007;
   }
   public static void math() {
      Supplier<double> s = Ready:getNumber;
      double d = s.get();
      System.out.println(d);
   }
}
```

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

57. Which statement about the following class is correct?

```
package robot;
import java.util.concurrent.*;
public class PassButter extends RecursiveTask<String> {  // j1
   final int remainder;
   public PassButter(int remainder) {  // j2
      this.remainder = remainder;
   }
   @Override
   protected String compute() {
      if (remainder <= 1)
         return "1";
      else {
         PassButter otherTask = new PassButter(remainder - 1);
         String otherValue = otherTask.fork().join();  // j3
         return otherValue
            + new PassButter(remainder - 2).compute();
      }
   }
   public static void main(String[] purpose) {
      ForkJoinPool pool = new ForkJoinPool();
      ForkJoinTask<?> task = new PassButter(10);
      System.out.print(pool.invoke(task));
      pool.shutdown();
   }
}
```

1. The code does not compile due to line `j1`.
2. The code does not compile due to line `j2`.
3. The code does not compile due to line `j3`.
4. The code compiles and properly implements the fork/join framework in a multi-threaded manner.
5. The code compiles but does not implement the fork/join framework in a proper multi-threaded manner.
6. The class compiles and prints an exception at runtime.

58. 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) {
      Stream<Boolean> hide = Stream.of(true, false, true);
      Predicate<Boolean> pred = b -> b;
      boolean found = hide.filter(pred).__________(pred);
      System.out.println(found);
   }
}
```

1. Only `anyMatch()`
2. Only `allMatch()`
3. Both `anyMatch()` and `allMatch()`
4. Only `noneMatch()`
5. The code does not compile with any of these options.

59. Given the following code, Java will try to find a matching resource bundle. Which order will Java search to find a match?

```
Locale.setDefault(new Locale("en"));
ResourceBundle.getBundle("AB", new Locale("fr"));
```

1. `AB.class`, `AB.properties`, `AB_en.properties`, `AB_fr.properties`
2. `AB.properties`, `AB.class`, `AB_en.properties`, `AB_fr.properties`
3. `AB_en.properties`, `AB_fr.properties`, `AB.class`, `AB.properties`
4. `AB_fr.properties`, `AB.class`, `AB.properties`, `AB_en.properties`
5. `AB_fr.properties`, `AB_en.properties`, `AB.class`, `AB.properties`
6. `AB_fr.properties`, `AB_en.properties`, `AB.properties`, `AB.class`

60. What is the result of the following?

```
Set<Integer> dice = new TreeSet<>();
dice.add(6);
dice.add(6);
dice.add(4);
dice.stream().filter(n -> n != 4).forEach(System.out::println).coun
```

1. It prints just one line.
2. It prints one line and then the number 3.
3. There is no output.
4. The code does not compile.

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

61. Given the following two property files in the pod package, what does the following class output?

```
pod.container.properties
name=generic
number=2
pod.container_en.properties
name=Docker
type=container
package pod;
import java.util.*;
public class WhatKind {
   public static void main(String[] args) {
      Locale.setDefault(new Locale("ja"));
      ResourceBundle rb = ResourceBundle.getBundle("pod.container")
      String name = rb.getString("name");    // r1
      String type = rb.getString("type");    // r2
      System.out.println(name + " " + type);   }
}
```

1. `Docker container`
2. `generic container`
3. `generic null`
4. The code does not compile.
5. Line `r1` throws an exception.
6. Line `r2` throws an exception.

62. What is the result of the following?

```
import java.util.stream.*;
public class StreamOfStreams {
   public static void main(String[] args) {
      Integer result =
         Stream.of(getNums(9, 8), getNums(22, 33))  // c1
         .filter(x -> !x.isEmpty())                 // c2
         .flatMap(x -> x)                           // c3
         .max((a, b) -> a - b)                      // c4
         .get();
      System.out.println(result);
   }
   private static Stream<Integer> getNums(int num1, int num2) {
      return Stream.of(num1, num2);
   }
}
```

1. The code compiles and outputs `8`.
2. The code compiles and outputs `33`.
3. The code does not compile due to line `c1`.
4. The code does not compile due to line `c2`.
5. The code does not compile due to line `c3`.
6. The code does not compile due to line `c4`.

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

1. `de`
2. `DE`
3. `de_DE`
4. `DE_de`

64. What is true of the following if the `music` database exists and contains a `songs` table with one row when run using a JDBC 4.0 driver? (Choose two.)

```
import java.sql.*;
public class Music {
   public static void main(String[] args) throws Exception {
      String url = "jdbc:derby:music";
      Connection conn = DriverManager.getConnection(url);
      Statement stmt = conn.createStatement();
      stmt.execute("update songs set name = 'The New Song'");
   }
}
```

1. The code does not compile.

2. The code does not update the database because it calls `execute()` rather than `executeUpdate()`.
3. The code does not update the database because the `Statement` is never closed.
4. The code runs without error.
5. The `execute()` method returns a `boolean`.
6. The `execute()` method returns an `int`.

65. How many of the following pairs of values can fill in the blanks to comply with the contract of the `hashCode()` and `equals()` methods?

```
class Sticker {
   @Override
   public int hashCode() {
      return _______________ ;
   }
   @Override
   public boolean equals(Sticker o) {
      return _______________;
   }
}
```

1. `5, false`
2. `5, true`
3. `new Random().nextInt(), false`
4. `new Random().nextInt(), true`

1. None
2. One
3. Two
4. Three
5. Four
6. None of the above. The code does not compile with any of the options.

66. What is the output of the following application?

```
package winter;

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 static void dress() {
      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();
   }
}
```

1. `Starting...Insulation:20`
2. `Starting...Insulation:40`
3. The code does not compile because of line `v1`.
4. The code does not compile because of line `v2`.
5. The code does not compile for a different reason.

67. Which statements about the following application are true?

```
1:  package armory;
2:  import java.util.function.*;
3:  class Shield {}
4:  public class Sword {
5:     public class Armor {
6:       int count;
7:        public final Function<Shield,Sword,Armor> dress = (h,w) -
8:       public final IntSupplier<Integer> addDragon = () -> count+
9:     }
10:    public static void main(String[] knight) {
11:      final Armor a = new Armor();
12:      a.dress.apply(new Shield(), new Sword());
13:      a.addDragon.getAsInt();
14:    }
15: }
```

1. The lambda expression for `dress` on line 7 compiles without issue.
2. The lambda expression for `addDragon` on line 8 compiles without issue.
3. Not counting the lambda expressions on lines 7 and 8, the code does not contain any compilation errors.

1. I only
2. I and II only
3. I, II, and III
4. II and III only
5. None of the above

68. Which two conditions best describe a thread that appears to be active but is perpetually stuck and never able to finish its task? (Choose two.)

1. Deadlock
2. Livelock
3. Loss of precision
4. Out of memory error
5. Race condition
6. Starvation

69. Which statements are true about the following date/times? (Choose two.)

```
2017-04-01T17:00+03:00[Africa/Nairobi]
2017-04-01T10:00-05:00[America/Panama]
```

1. The first date/time is earlier.
2. The second date/time is earlier.
3. Both represent the same date/time.
4. The two date/times are zero hours apart.
5. The two date/times are one hour apart.
6. The two date/times are two hours apart.

70. 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
  }
}
```

1. The code doesn't compile due to line `c1`.
2. The code doesn't compile due to line `c2`.
3. The code doesn't compile due to line `c3`.
4. The code compiles and runs without output.
5. The code compiles but throws an exception at runtime.

71. What is true about the following code? (Choose two.)

```
public static void main(String[] args) throws Exception {
   String url = "jdbc:derby:hats;create=true";
   Connection conn = null;
   Statement stmt = null;
   try {
     conn = DriverManager.getConnection(url);
     stmt = conn.createStatement();
     stmt.executeUpdate(        "CREATE TABLE caps (name varchar(25
   } finally {
      conn.close();
      stmt.close();
   }
}
```

1. If using a JDBC 3.0 driver, this code throws an exception.
2. If using a JDBC 4.0 driver, this code throws an exception.
3. The resources are closed in the wrong order.
4. The resources are closed in the right order.
5. The `Connection` is created incorrectly.
6. The `Statement` is created incorrectly.

72. How many lines of the following application contain a compilation error?

```
package puzzle;
final interface Finder {
   default long find() {return 20;}
}
abstract class Wanda {
   abstract long find();
}
final class Waldo extends Wanda implements Finder {
   long find() {return 40;}
   public static final void main(String[] pictures) {
      final Finder f = new Waldo();
      System.out.print(f.find());
   }
}
```

1. One
2. Two
3. Three
4. None. The code compiles and prints `20` at runtime.
5. None. The code compiles and prints `40` at runtime.

73. What is the output of the following?

```
package reader;
import java.util.stream.*;

public class Books {
   public static void main(String[] args) {
      IntStream pages = IntStream.of(200, 300);
      long total = pages.sum();
      long count = pages.count();
      System.out.println(total + "-" + count);
   }
}
```

1. `2-2`
2. `200-1`
3. `500-0`
4. `500-2`
5. The code does not compile.
6. The code compiles but throws an exception at runtime.

74. What is the output of executing the following code snippet?

```
ExecutorService e = Executors.newSingleThreadExecutor();
Runnable r1 = () -> Stream.of(1,2,3).parallel();
Callable r2 = () -> Stream.of(4,5,6).parallel();

Future<Stream> f1 = e.submit(r1);
Future<Stream> f2 = e.submit(r2);

Stream<Integer> s = Stream.of(f1.get(),f2.get())
      .flatMap(p -> p)
      .parallelStream();

ConcurrentMap<Boolean,List<Integer>> r =
      s.collect(Collectors.groupingByConcurrent(i -> i%2==0));
System.out.println(r.get(false).size()+" "+r.get(true).size());
```

1. `3 3`
2. `2 4`
3. The code does not compile due to one error.
4. The code does not compile due to two errors.
5. The code does not compile due to three errors.
6. The code compiles but a `NullPointerException` is thrown at runtime.

75. Fill in the blanks: If your application is___________ , it must first have been ___________ with respect to supporting multiple languages.

1. extracted, internationalized
2. extracted, localized
3. internationalized, extracted
4. internationalized, localized
5. localized, extracted
6. localized, internationalized

76. Which statement about the following class is true? Assume the file system is available and able to be modified.

```
package forest;
import java.io.File;
public class CreateTree {
   public boolean createTree(String tree) {
      if(new File(tree).exists()) {
         return true;
      } else {
         return new File(tree).mkdir();
      }
   }
   public static void main(String[] seeds) {
      final CreateTree creator = new CreateTree();
      System.out.print(creator.createTree("/woods/forest"));
   }
}
```

1. The class compiles and always prints `true` at runtime.
2. The class compiles and always prints `false` at runtime.
3. The class compiles but the output cannot be determined until runtime.
4. The class compiles but may throw an exception at runtime.
5. The class does not compile.

77. What does the following print?

```
 class SmartWatch extends Watch {
   private String getType() { return "smart watch"; }
   public String getName() {
      return getType() + ",";
   }
}
public class Watch {
   private String getType() { return "watch"; }
   public String getName(String suffix) {
      return getType() + suffix;
   }
   public static void main(String[] args) {
      Watch watch = new Watch();
      Watch smartWatch = new SmartWatch();
      System.out.print(watch.getName(","));
      System.out.print(smartWatch.getName(""));
   }
}
```

1. `smart watch,smart watch`
2. `smart watch,watch`
3. `watch,smart watch`
4. `watch,watch`
5. None of the above

78. In most of the United States, daylight savings time ends on November 5, 2017 at 02:00 a.m., and we repeat that hour. What is the output of the following?

```
import java.time.*;

public class FallBack {

   public static void main(String[] args) {
      LocalDate localDate = LocalDate.of(2017, Month.NOVEMBER, 5);
      LocalTime localTime = LocalTime.of(1, 0);
      ZoneId zone = ZoneId.of("America/New_York");
      ZonedDateTime z = ZonedDateTime.of(localDate, localTime, zone

      for (int i = 0; i < 6; i++)
         z = z.plusHours(1);

      System.out.println(z.getHour());
   }
}
```

1. 5
2. 6
3. 7
4. The code does not compile.
5. The code compiles but throws an exception at runtime.

79. Which statements about the following application are true?

```
package party;
import java.util.concurrent.*;
public class Plan {
   private ExecutorService service = Executors.newCachedThreadPool(
   public void planEvents() {
      service.scheduleWithFixedDelay(
            () -> System.out.print("Check food stock"),
            1, TimeUnit.HOURS);
      service.scheduleAtFixedRate(
            () -> System.out.print("Check drink stock"),
            1, 1000, TimeUnit.SECONDS);
      service.execute(() -> System.out.print("Take out trash"));
   }
}
```

1. The `scheduleWithFixedDelay()` method call compiles.
2. The `scheduleAtFixedRate()` method call compiles.
3. The `execute()` method call compiles.

1. I only
2. II only
3. III only
4. I and II
5. I, II, and III
6. None of the above

80. Which of the following classes are checked exception? (Choose three.)

1. `java.io.NotSerializableException`
2. `java.lang.AssertionError`
3. `java.lang.IllegalArgumentException`
4. `java.sql.SQLException`
5. `java.text.ParseException`
6. `java.util.MissingResourceException`

81. Which of the following are valid functional interfaces? (Choose two.)

1. `interface CanClimb {default void climb() {}`
1. `static void climb(int x) {}}`
2. `interface CanDance {int dance() { return 5;}}`
3. `interface CanFly {abstract void fly();}`
4. `interface CanRun {void run();`
1. `static double runFaster() {return 2.0;}}`
5. `interface CanSwim {abstract Long swim();`

1. `boolean test();}`

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

1. `jdbc;box;magic`
2. `jdbc;magic;@127.0.0.1:1234`
3. `jdbc;magic;//@127.0.0.1:1234`
4. `jdbc;magic;127.0.0.1:1234/box`
5. `magic;jdbc;127.0.0.1:1234/box`

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

83. What is the output of the following?

```
Stream<String> s = Stream.of("speak", "bark", "meow", "growl");
Map<Integer, String> map = s.collect(toMap(String::length, k -> k))
System.out.println(map.size() + " " + map.get(4));
```

1. `2 bark`
2. `2 meow`
3. `4 bark`
4. `4 meow`
5. The output is not guaranteed.
6. The code compiles but throws an exception at runtime.

84. 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;}
}
class ElectricBass implements DoubleBass, BassGuitar {
   @Override public void strum() {System.out.print("A");}
}
public class RockBand {
   public static void main(String[] strings) {
      final class MyElectricBass extends ElectricBass {
         public void strum() {System.out.print("E");}
      }
   }
}
```

1. `A`
2. `E`
3. The code compiles and runs without issue but does not print anything.
4. One line of code does not compile.
5. Two lines of code do not compile.
6. Three lines of code do not compile.

85. Which NIO.2 `Files` methods return a `Stream`? (Choose three.)

1. `find()`
2. `lines()`
3. `list()`
4. `listFiles()`
5. `readAllLines()`

6. `walkFileTree()`

PREV
Chapter 22 Localization

NEXT
Appendix Answers to Review Questions