

Multiple-Choice Quiz – Week 13

Exceptions and File I/O · for reference: Weeks 9–10

Question 1. Which keyword do you use in Java to signal that an error condition has occurred and abort normal execution?

- ☐ A) catch
- ☐ B) throw
- ☐ C) raise
- ☐ D) error

Question 2. What is the common superclass that both Exception and Error extend, at the root of Java's exception hierarchy?

- ☐ A) Throwable
- ☐ B) Object
- ☐ C) RuntimeException
- ☐ D) Class

Question 3. A method calls `Files.readString(path)` without wrapping the call in a try-catch or adding a throws clause – what happens?

- ☐ A) It compiles, and the exception is silently ignored at runtime
- ☐ B) It compiles, but throws a `NullPointerException` when called
- ☐ C) It fails to compile
- ☐ D) It compiles, and the file is read successfully regardless

Question 4. Which of these correctly declares a new checked exception type in Java?

- ☐ A) `class LowBatteryException extends RuntimeException { }`
- ☐ B) `class LowBatteryException extends Exception { }`
- ☐ C) `class LowBatteryException implements Exception { }`
- ☐ D) `checked class LowBatteryException { }`

Question 5. True or false: a catch (`IllegalArgumentException e`) block also catches a thrown `NumberFormatException`.

- ☐ A) True
- ☐ B) False

Question 6. What does try-with-resources automatically provide?

- ☐ A) It closes the resource automatically, even if an exception was thrown
- ☐ B) It skips exception handling entirely
- ☐ C) It only closes the resource if no exception was thrown
- ☐ D) It ensures that no exceptions happen

Question 7. When does the code inside a finally block run?

- ☐ A) Always, whether or not an exception was thrown or caught
- ☐ B) Only when no exception was thrown
- ☐ C) Only when a thrown exception is caught
- ☐ D) Only when the method returns normally

Question 8. Which Files method reads an entire text file into a single String?

- ☐ A) `Files.newBufferedReader(path)`
- ☐ B) `Files.readString(path)`

- ☐ C) `Files.write(path, content)`
- ☐ D) `Files.readAllLines(path)`

Question 9. Calling `s.trim()` on a `String s` returns a new trimmed string rather than changing `s` itself – what property of `String` does this demonstrate?

- ☐ A) Immutability
- ☐ B) Encapsulation
- ☐ C) Static binding
- ☐ D) Aliasing

Question 10. Which keyword marks a field so its value cannot be reassigned after initialization?

- ☐ A) `final`
- ☐ B) `static`
- ☐ C) `private`
- ☐ D) `this`

Question 11. Which operator checks at runtime whether an object is an instance of a particular class or subclass?

- ☐ A) `instanceof`
- ☐ B) `typeof`
- ☐ C) `is`
- ☐ D) `getClass()`

Question 12. Which keyword does a class use to declare that it provides every method promised by an interface?

- ☐ A) `extends`
- ☐ B) `interfaces`
- ☐ C) `inherits`
- ☐ D) `implements`

Answer Key — Week 13 Quiz

Concept tested in *italics*.

- Q1 B** *Throwing an exception.* `throw` signals an error and aborts normal execution. `catch` handles an already-thrown exception; `raise` and `error` are not Java keywords.
- Q2 A** *The exception hierarchy.* `Throwable` sits at the root, with both `Exception` and `Error` extending it directly. `Object` is the root of all Java classes, not specifically exceptions; `RuntimeException` is only a branch of `Exception`; `Class` is unrelated.
- Q3 C** *Checked exceptions must be handled.* `IOException` is checked, so a method that doesn't catch it must declare `throws IOException` — omitting both is a compile error, not a silent or delayed failure.
- Q4 B** *User-defined exceptions.* Extending `Exception` makes a checked exception. Extending `RuntimeException` (A) makes it unchecked instead; `Exception` is a class so it cannot be implemented (C); `checked` is not a Java keyword (D).
- Q5 A** *Try-catch uses subtyping.* `NumberFormatException` is a subclass of `IllegalArgumentException`, so a `catch (IllegalArgumentException e)` block matches it too.
- Q6 A** *Try-with-resources.* It closes the declared resource(s) automatically when the block exits, exception or not, without a manual `finally`. It does not skip exception handling, does not depend on success, and closing happens synchronously, not on a background thread.
- Q7 A** *The finally block.* Code in `finally` always runs after the `try` and any `catch` blocks, regardless of whether an exception was thrown or caught.
- Q8 B** *File I/O methods.* `Files.readString(path)` returns the whole file as one `String`. `readAllLines` returns a `List` of lines instead, `newBufferedReader` returns a reader for line-by-line processing, and `write` is for writing, not reading.
- Q9 A** *Immutability (Week 9 review).* `String` is immutable, so methods like `trim()` return a new `String` rather than modifying the original.
- Q10 A** *The final keyword (Week 9 review).* A `final` field's value can be set once and never reassigned. `static` marks class-level sharing, `private` controls access, and this refers to the current object — none of them relate to reassignment.
- Q11 A** *The instanceof operator (Week 10 review).* `instanceof` checks an object's runtime type against a class or subclass. `typeof` and `is` are not Java keywords; `getClass()` returns a `Class` object rather than performing the check itself.
- Q12 D** *Interfaces (Week 10 review).* `implements` is the keyword a class uses to promise it provides every method an interface declares. `extends` is for subclassing another class; `interfaces` and `inherits` are not Java keywords.