

Multiple-Choice Quiz — Week 4

Arrays · for reference: Weeks 1–3

Question 1. At what index does the first element of a Java array sit?

- ☐ A) 0
- ☐ B) 1
- ☐ C) 10
- ☐ D) 16

Question 2. Can a Java array be resized after creation?

- ☐ A) Yes — it happens automatically
- ☐ B) Yes — you do it manually
- ☐ C) No — the size is fixed
- ☐ D) No — size can only decrease

Question 3. Given `int[] scores = {70, 85, 90, 60};`, what does `scores[2]` print?

- ☐ A) 70
- ☐ B) 85
- ☐ C) 90
- ☐ D) 60

Question 4. A programmer wants to declare a constant holding the maximum allowed number of students. Which name follows Java conventions?

- ☐ A) `maxStudents`
- ☐ B) `MaxStudents`
- ☐ C) `max_students`
- ☐ D) `MAX_STUDENTS`

Question 5. What is the result of `9 / 5` in Java?

- ☐ A) 1
- ☐ B) 1.8
- ☐ C) 2
- ☐ D) 4

Question 6. What does the following program print?

```
double[] a = new double[4];
System.out.println(a[2]);
```

- ☐ A) `null`
- ☐ B) `4.0`
- ☐ C) `0.0`
- ☐ D) Nothing — it does not compile

Question 7. What does Java produce when you divide a double by zero, as in `1.0 / 0.0`?

- ☐ A) An `ArithmeticException` — just like dividing an int by zero
- ☐ B) `NaN` (not a number)
- ☐ C) Infinity
- ☐ D) A compile error — division by zero is always illegal in Java

Question 8. Which expression correctly accesses the last element of `int[] a`?

- ☐ A) `a[a.length]`
- ☐ B) `a[a.length - 1]`
- ☐ C) `a[-1]`
- ☐ D) `a[a.size() - 1]`

Question 9. `sum` and `count` are both `int`. Which expression gives a decimal result?

- ☐ A) `sum / count`
- ☐ B) `sum % count`
- ☐ C) `(double) sum / count`
- ☐ D) Java returns a decimal automatically when the result is not a whole number

Question 10. Which type of error does Java always detect **before** the program runs?

- ☐ A) Accessing an out-of-bounds array index
- ☐ B) Calling a method on a null reference
- ☐ C) Assigning a String value to an int variable
- ☐ D) Dividing an integer by zero

Question 11. Two students store their quiz answers in separate arrays with identical contents. Which expression correctly checks whether the two arrays contain the same values in the same order?

- ☐ A) `alice == bob`
- ☐ B) `alice.equals(bob)`
- ☐ C) `Arrays.equals(alice, bob)`
- ☐ D) `Arrays.compare(alice, bob)`

Question 12. A for-loop declares its counter in the loop header: `for (int i = 0; i < n; i++) { ... }`. What happens if you try to use `i` after the closing `}`?

- ☐ A) `i` retains its last value and can be read normally
- ☐ B) `i` is automatically reset to 0 after the loop finishes
- ☐ C) The compiler reports an error — `i` is out of scope
- ☐ D) `i` becomes null once the loop ends

Answer Key — Week 4 Quiz

Concept tested in *italics*.

- Q1 A** *Zero-based indexing.* Java arrays are zero-indexed: the first element is always at index 0, and the last is at index `a.length - 1`.
- Q2 C** *Fixed size.* A Java array's size is set when it is created and cannot change. To work with more elements you must allocate a new, larger array (e.g. with `Arrays.copyOf`) and copy the data across.
- Q3 C** *Array indexing.* Indices start at 0, so `scores[2]` refers to the third element: 90. The other options are the values sitting at indices 1 and 3.
- Q4 D** *Naming conventions.* Constants use `SCREAMING_SNAKE_CASE`. `lowerCamelCase` is for variables, `UpperCamelCase` for classes, and `snake_case` is not a Java convention.
- Q5 A** *Integer division.* Both 9 and 5 are `int`, so `/` keeps only the quotient: $9 / 5 = 1$. The remainder (4) is discarded. Use `9.0 / 5` or a cast to get 1.8.
- Q6 C** *Default values for numeric arrays.* `new double[4]` initializes all elements to `0.0`.
- Q7 C** *double division by zero.* Unlike integer division, `double / 0.0` does not throw — Java returns the special value `Infinity`. `0.0 / 0.0` gives `NaN`.
- Q8 B** *Last element access.* The last valid index is `a.length - 1` (zero-based indexing). `a[a.length]` is off-by-one and throws `AIOOBE`; `a[-1]` is not valid in Java; arrays have no `size()` method.
- Q9 C** *Integer division.* When both operands are `int`, `/` discards the decimal part. Casting one operand to `double` before dividing forces floating-point division. `sum % count` gives the remainder, not the average.
- Q10 C** *Compile-time vs. runtime errors.* Type mismatches (like assigning a `String` to an `int`) are caught by `javac` before the program runs. The other three all compile fine but crash at runtime: `AIOOBE`, `NullPointerException`, and `ArithmeticException` respectively.
- Q11 C** *Content equality.* `==` and `.equals()` both compare references (same object in memory). `Arrays.compare` returns an ordering integer, not a yes/no. Only `Arrays.equals` compares element by element.
- Q12 C** *For-loop variable scope.* A variable declared in the `for` header is scoped to that loop's block. Trying to read `i` after `}` is a compile-time error: "cannot find symbol".