
The free-response section is graded against a rubric of specific code behaviours. Most lost points are off-by-one bounds, missing null or empty checks, and methods that print instead of returning. Trace your code on the boundary cases before moving on. 简答题按代码行为给分, 失分多为边界越界、缺少空值判断, 以及该返回却打印。

Loops and arrays

Looping to `<= length`. Valid indices run from 0 to `length - 1`; `<=` throws `ArrayIndexOutOfBoundsException`.

Using `.length()` on an array or `.length` on a String. Arrays use the FIELD `length`; String and ArrayList use methods.

Removing from an ArrayList inside a forward loop, which skips the next element. Iterate backwards or adjust the index.

Assigning to the variable of an enhanced for loop and expecting the array to change. It does not.

Forgetting that a 2-D array's first index is the ROW; row and column reversed is the standard trap.

Methods and returns

Printing the answer instead of returning it. If the signature says `return`, printing scores nothing.

Omitting a return on some path through the method, so not every branch produces a value.

Changing the given method signature. The rubric checks it, and an altered signature can void the whole part.

Rewriting a helper method the question told you to assume. Call it; do not reimplement it.

Ignoring the precondition, and adding checks the question said were unnecessary – harmless, but it costs time.

Objects and correctness

Comparing Strings or objects with `==`, which compares references. Use `.equals`.

Calling a method on a reference that could be null without checking it first.

Using integer division by accident: `7/2` is 3, and one operand must be a double to get 3.5.

Forgetting that a String is immutable, so `s.substring(...)` must be assigned to something.

Writing a recursive method with no reachable base case.

Assuming a subclass method is called when the reference type is the superclass – it IS, because overriding resolves at run time; the reverse assumption loses marks in tracing questions.