

1Z0-808 Exam Topic Reviewer

TopicId: 1009

Topic: Conditional Statements (if/else, switch)

August 5, 2025

Topic 1009: Conditional Statements (if/else, switch)

Thinking Like the Compiler: Making Decisions

Program flow isn't always linear. Conditional statements allow a program to make decisions and execute different blocks of code. The exam will test your knowledge of the strict syntax rules for these statements and the logical flow they create, especially in oddly formatted code or tricky 'switch' blocks.

The if-else Construct

This is the most fundamental decision-making tool.

The Condition Must Be a Boolean

This is a non-negotiable rule in Java. The expression inside the `if(...)` parentheses must evaluate to either `true` or `false`.

```
int x = 10;
if (x = 5) { ... } // COMPILER ERROR! An assignment (x=5) results in an int,
                  // not a boolean.
```

```
if (x == 10) { ... } // CORRECT. The result of == is a boolean.
```

Optional Braces { } and the Dangling else

The exam loves to test code where the curly braces are omitted. An `if` or `else` block without braces can only contain a **single statement**.

An `else` clause always binds to the nearest preceding 'if' that doesn't have an 'else' yet. This is known as the "dangling else" problem.

```
int score = 80;
boolean isGraded = false;

// Which 'if' does this 'else' belong to?
if (score > 60)
    if (isGraded)
        System.out.println("Pass");
else // This 'else' belongs to 'if (isGraded)', NOT 'if (score > 60)'
    System.out.println("Not graded yet");

// Correctly formatted with braces for clarity:
if (score > 60) {
    if (isGraded) {
        System.out.println("Pass");
    } else {
        System.out.println("Not graded yet");
    }
}
```

The Ternary Operator (? :)

This is a compact shorthand for an ‘if-else’ statement that produces a value.

Syntax: `booleanExpression ? valueIfTrue : valueIfFalse;`

```
int score = 75;
String result;
if (score > 60) {
    result = "Pass";
} else {
    result = "Fail";
}
```

```
// The same logic using a ternary operator:
String ternaryResult = score > 60 ? "Pass" : "Fail";
```

The switch Statement

This is a huge topic for the exam. You must know its rules inside and out.

Valid switch Argument Types

The variable you ‘switch’ on must be of a compatible type. For Java 8, these are:

- Primitives: `byte`, `short`, `char`, `int`.
- Their wrapper classes: `Byte`, `Short`, `Character`, `Integer`.
- `String` (since Java 7).
- `enum` types.

Crucial Exam Tip: You **cannot** switch on a `long`, `float`, `double`, or `boolean`.

case Values Must Be Compile-Time Constants

The value for each ‘case’ must be known at compile time. This means it must be a literal or a `final` variable.

```
final int MONDAY = 1;
int day = 2;
switch (day) {
    case MONDAY: System.out.println("Work"); break; // OK, MONDAY is final
    case 2: System.out.println("Work"); break;      // OK, 2 is a literal
    // case day: System.out.println("Huh?"); break; // COMPILE ERROR! 'day' is not final
}
```

The Fall-Through Trap

This is the most tested aspect of ‘switch’ statements. If a `case` block does not end with a `break`, execution **falls through** to the next `case` block and executes its statements, continuing until a `break` is found or the `switch` ends.

```
int option = 2;
switch (option) {
    case 1:
        System.out.print("A");
    case 2: // Matches here, execution starts
        System.out.print("B"); // Prints B. No break, so it falls through.
    case 3:
        System.out.print("C"); // Prints C. No break, so it falls through.
    default:
        System.out.print("D"); // Prints D.
}
```

// FINAL OUTPUT: BCD

The `default` block is optional and can be placed anywhere, but its location affects the fall-through logic.

Key Takeaways for the 1Z0-808 Exam

- **if needs a boolean:** An assignment like `if(x=5)` is a compile error.
- **Beware of Missing Braces:** Track the logic of single-statement `if/else` blocks carefully and know the dangling `else` rule.
- **Memorize switch types:** Know which types are allowed (`int`, `String`, `char`, etc.) and which are forbidden (`long`, `double`, `boolean`).
- **Master Fall-Through:** When you see a `switch` statement, your first action should be to look for the `break` statements and trace the execution path.