

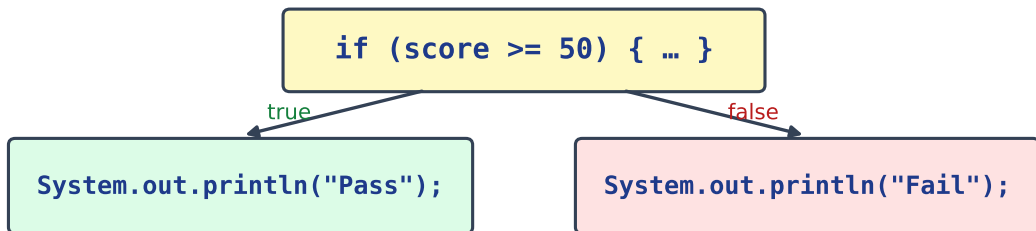
Booleans & selection

Java Reference

if / else

`if` runs a block when a condition is true; `else if` and `else` add more cases. The condition goes in (), the block in { }.

```
public class Main {  
    public static void main(String[] args) {  
        int score = 72;  
        if (score >= 80) {  
            System.out.println("A");  
        } else if (score >= 60) {  
            System.out.println("B");  
        } else {  
            System.out.println("fail");  
        }  
    }  
}
```



Exam tip: always use braces; else if chains more cases; only one branch runs

if chooses the true branch; else the false branch

Logical operators & comparisons

Compare with `==`, `!=`, `<`, `>`, `<=`, `>=` — a comparison gives a `boolean`. Combine with `&&` (and), `||` (or), `!` (not) — the logical operators. For `Strings`, use `.equals(...)`, `not ==`.

```
public class Main {
    public static void main(String[] args) {
        int age = 16;
        boolean member = true;
        System.out.println(age >= 18 && member);    // false
        String a = "hi";
        System.out.println(a.equals("hi"));        // true
    }
}
```

switch

`switch` chooses among many fixed values. Each case ends with `break`; default is the fallback.

```
public class Main {
    public static void main(String[] args) {
        int day = 3;
        switch (day) {
            case 1: System.out.println("Mon"); break;
            case 3: System.out.println("Wed"); break;
            default: System.out.println("other");
        }
    }
}
```

Common mistakes

- A condition must be a `boolean`; `if (x = 5)` does not compile (use `==`).
- Each `switch` case needs a `break`; or control falls through to the next case.
- `&&` and `||` are the logical operators; `&` and `|` are bitwise.