

Branches on Branches

Nested if statements and else if chains



A question inside a question

PAUSE AND TRY

What does this plan say on a sunny Tuesday? A rainy Sunday?

HOPPER'S DAY PLAN

If it is the weekend:

if it is sunny, go to the park;

otherwise, watch a movie.

Otherwise, go to school.

Talk with a partner. On which days does the plan ask about the weather at all?

By the end of class you can...

- 1 Trace an if or if-else inside another one, and say which conditions Java checks.
- 2 Explain why an inner condition is checked only inside its branch, and use that as a guard.
- 3 Work out which if an else belongs to, and use braces to pair them the way you mean.
- 4 Trace an if, else if, else chain and explain why at most one branch runs.
- 5 Order the conditions in a chain, and decide when a trailing else is needed.

1

PART 1

An if inside an if

The inner if lives in the outer body

LIBRARY CHECK

```
int age = 15;
boolean hasCard = true;
if (age >= 13) {
    System.out.println("Teen section");
    if (hasCard) {
        System.out.println("You can borrow");
    }
}
System.out.println("Welcome");
```

OUTPUT

```
Teen section
You can borrow
Welcome
```



A **nested if** is an if inside another if's body. Its condition is checked **only when the outer condition is true**.

If age is...	Checks hasCard?
15	Yes
10	No: whole body skipped

A guard keeps a risky check safe

GUARD OUTSIDE

```
int players = 0;
int points = 0;
if (players > 0) {
    if (points / players > 10) {
        System.out.println("Strong team");
    }
}
System.out.println("Done");
```

OUTPUT

Done

Step	What Java does
1	players > 0 -> 0 > 0 -> false
2	Skips the outer body. No division.
3	Prints Done.



A **guard** is an outer condition that makes sure the inner check is safe to run.

A guard only protects what is inside it



Put the safety check **outside**. An inner guard is too late: the outer condition has already run.

GUARD INSIDE: CRASHES

```
int players = 0;
int points = 0;
if (points / players > 10) {
    if (players > 0) {
        System.out.println("Strong team");
    }
}
System.out.println("Done");
```

PROGRAM STOPS

```
Exception in thread "main"
java.lang.ArithmeticException:
/ by zero
```

Nothing prints, not even Done.

A large white square with rounded corners, tilted slightly to the right, containing the number '2' in a bold, dark teal font.

2

PART 2

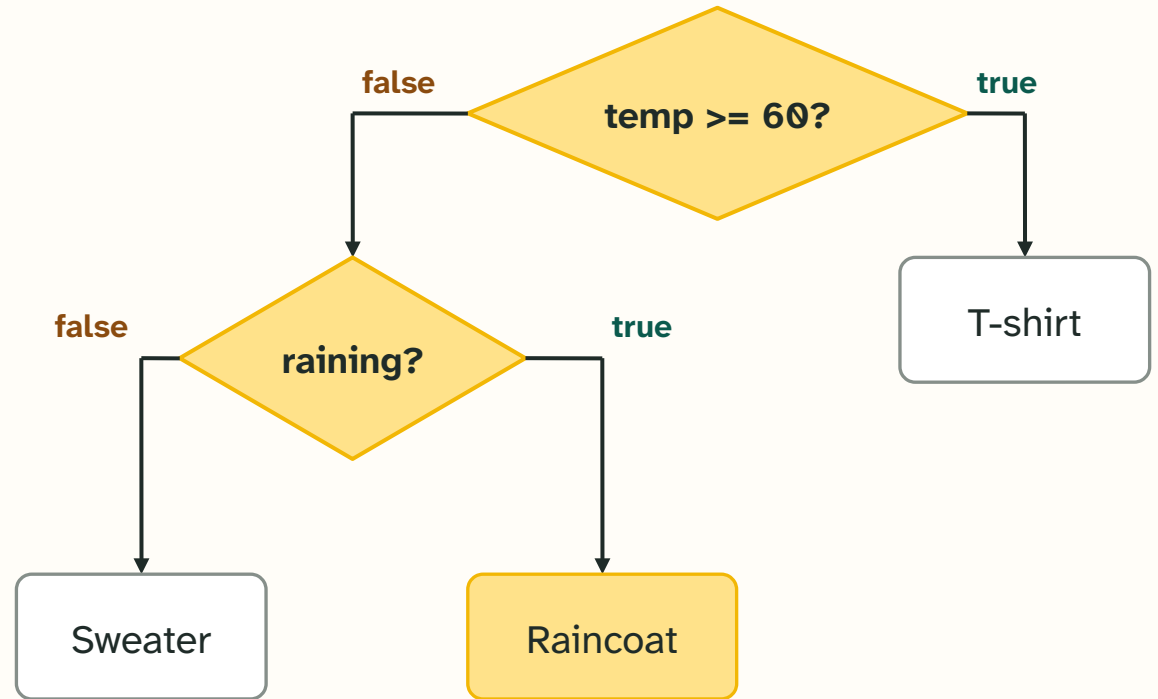
Which branch, which else?

Nesting in the else branch

```
int temp = 40;
boolean raining = true;
if (temp >= 60) {
    System.out.println("T-shirt");
} else {
    if (raining) {
        System.out.println("Raincoat");
    } else {
        System.out.println("Sweater");
    }
}
```

OUTPUT

Raincoat



One path runs. Three possible results.

An inner else answers only its own if



An `else` is not "otherwise, for anything that failed." Each `else` answers **exactly one if**.

```
int age = 14;
boolean hasPass = false;
if (age >= 16) {
    if (hasPass) {
        System.out.println("Drive on");
    } else {
        System.out.println("Get a pass first");
    }
}
System.out.println("Done");
```

OUTPUT

Done

The inner `else` runs only when `age >= 16` is true **and** `hasPass` is false.

Without braces, an else finds the nearest if



No braces? An `else` pairs with the **nearest earlier if without an `else`**, whatever the indentation says.

LOOKS LIKE IT PAIRS WITH THE OUTER IF

```
int n = 3;
if (n > 0)
    if (n > 5)
        System.out.println("big");
else
    System.out.println("not positive?");
```

OUTPUT

not positive?

BRACES MAKE IT MEAN WHAT IT SAYS

```
int n = 3;
if (n > 0) {
    if (n > 5) {
        System.out.println("big");
    }
} else {
    System.out.println("not positive");
}
System.out.println("end");
```

OUTPUT

end

Ready for the game?

PAUSE AND TRY

What prints for each player?

```
if (hours >= 8) {  
    if (rested) {  
        System.out.println("Ready");  
    } else {  
        System.out.println("Nap first");  
    }  
} else {  
    System.out.println("Sleep more");  
}
```

hours	rested	Prints
9	true	
9	false	
6	true	

3

PART 3

Multiway selection: else if

The first true condition wins

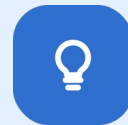
SPEED CHECK

```
int speed = 72;  
if (speed > 80) {  
    System.out.println("Ticket");  
} else if (speed > 65) {  
    System.out.println("Warning");  
} else {  
    System.out.println("Safe");  
}
```

OUTPUT

Warning

Check	Result	Then
72 > 80	false	go on
72 > 65	true	run, then stop
else	-	skipped



In a **multiway selection**, at most one body runs: the one for the **first** true condition.

Order matters in a chain



A broad condition first **swallows** every value meant for the branches below it.

CONDITIONS IN THE WRONG ORDER

```
int speed = 90;
if (speed > 65) {
    System.out.println("Warning");
} else if (speed > 80) {
    System.out.println("Ticket");
} else {
    System.out.println("Safe");
}
```

OUTPUT

Warning

90 is over 80, but it never gets that far. Ticket is **unreachable**: any speed over 80 is also over 65.

The trailing else catches the rest

NO TRAILING ELSE

```
int coins = 3;
if (coins >= 100) {
    System.out.println("Gold chest");
} else if (coins >= 50) {
    System.out.println("Silver chest");
}
System.out.println("Coins: " + coins);
```

OUTPUT

Coins: 3

WITH A TRAILING ELSE

```
int coins = 3;
if (coins >= 100) {
    System.out.println("Gold chest");
} else if (coins >= 50) {
    System.out.println("Silver chest");
} else {
    System.out.println("No chest");
}
System.out.println("Coins: " + coins);
```

OUTPUT

No chest
Coins: 3

The plain else goes last



A plain `else` ends the chain. Nothing can be attached after it, and it never takes a condition.

DOES NOT COMPILE

```
int coins = 3;
if (coins >= 100) {
    System.out.println("Gold chest");
} else {
    System.out.println("No chest");
} else if (coins >= 50) {
    System.out.println("Silver chest");
}
```

COMPILER SAYS

error: 'else' without 'if'

Fix: move the plain `else` and its body to the end of the chain.

Shipping tiers

PAUSE AND TRY

What prints for each weight?

```
double cost;  
if (pounds > 20) {  
    cost = 25.0;  
} else if (pounds > 5) {  
    cost = 12.0;  
} else {  
    cost = 6.0;  
}  
System.out.println(cost);
```

pounds	Prints
25	
20	
5	
1	

Write the chain

PAUSE AND TRY

Write one if, else if, else chain that prints the right word for an int temp.

temp	Prints
below 0	Freezing
0 to 15	Cold
16 to 25	Mild
above 25	Hot

YOUR CODE

Hint: put the conditions in an order where each one only needs one comparison.

Words to know

Nested if

An if statement inside the body of another if or else.

Guard

An outer condition that makes an inner check safe to run.

Dangling else

With no braces, an else pairs with the nearest unpaired if.

Multiway selection

An if, else if, else chain. At most one body runs.

else if

Checked only when every condition above it was false.

Trailing else

Runs when no condition in the chain was true.

Nest or chain?



Explain to your partner when you would nest one if inside another, and when you would use an else if chain.

SENTENCE STARTER

An inner condition is checked only when _____, but in a chain Java runs the body of the _____ true condition.

Next: the Branches on Branches puzzle worksheet.