Lesson 3

1

Why Code Needs Choices

A Fundamental Question

How does a GPS choose 'Turn left' over 'Turn right'?

Programs must respond to different inputs dynamically. Without choices, code would only ever do one thing, every time.

2

The `if` Statement

The Basic Tool

Conditional Operators

- == (Is equal to)
- != (Is not equal to)
- > (Is greater than)
- < (Is less than)
- >= (Is greater than or equal to)
- <= (Is less than or equal to)

Conditional Statement

A structure that performs different actions depending on whether a specified condition is true or false.

```
if (condition) { //
statements when
<mark>condition is
true</mark> }
```

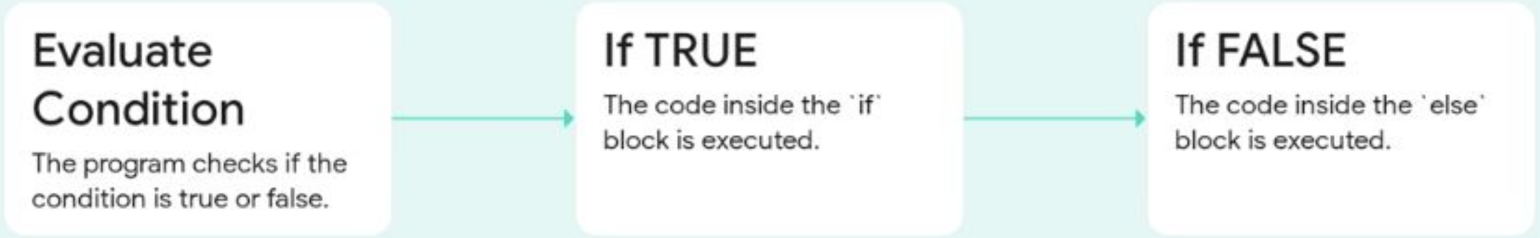
This code checks if `x` and `y` are equal. The message is printed only if the condition is true.

3

The `else` Alternative

Handling False Conditions

Logic of a Choice



Evaluate Condition

The program checks if the condition is true or false.

If TRUE

The code inside the `if` block is executed.

If FALSE

The code inside the `else` block is executed.

➡️

`if (score >= 60) { cout << "Pass"; }` A path for one outcome.

↔️

`... else { cout << "Fail"; }` A path for every outcome.

You can chain conditions to create a ladder of decisions. The code checks each one in order until one is true.

4

Complex Decisions

Combining Conditions

Logical Operators

Operators used to combine multiple conditions into one true/false result, like AND (&&) and OR (||).

Condition A	Condition B	A && B (AND)	A || B (OR)
true	true	true	true
true	false	false	true
false	true	false	true
false	false	false	false

Check if x is divisible by 2 AND y is divisible by 4: `(x%2==0) && (y%4==0)`

5

Code in Action

A Practical Example

Eligible to Graduate?

- `credits >= 60`
- `gpa >= 2.0`
- `holds == 0` (no holds)
- `courseReq == 0` (finished)

```
int credits; double gpa; int holds; int courseReq;
```

Setup variables to hold student data.

A single `if` statement uses `&&` to check if all four conditions are met at once.

The `else` block uses nested `if` statements to give the user specific, helpful feedback if they don't qualify.

Key Takeaways

- `if` statements check if a condition is true.
- `else` provides an alternative for false conditions.
- Logical operators (`&&`, `||`) combine checks.
- This allows for dynamic and responsive code.

Now you can make code choose, what decisions will you make?

Homework #2

FizzBuzz

Interactive FizzBuzz Homework

- Write a C++ program for FizzBuzz
- Take a single number from the user
- Apply FizzBuzz rules using conditional statements
- Practice std::cin for user input and conditionals