

Control Flow

◇ 1. **if** Statement

☑ Analogy:

You check **if it's raining**. If yes, 🌧️ **take umbrella**. If not, just walk normally — **do nothing extra**.

🔧 Code:

```
#include <iostream>
using namespace std;

int main() {
    int temp = 35;

    if (temp > 30) {
        cout << "It's hot! 🌞" << endl;
    }

    return 0;
}
```

🔍 Output:

It's hot! 🌞

📄 Flow:

Check condition → true → run if block
→ false → skip block

◇ 2. **if-else** Statement

☑ Analogy:

If it's raining 🌧️ → take umbrella **Else** ☀️ → wear sunglasses 😎

🔧 Code:

```
#include <iostream>
using namespace std;
```

```
int main() {  
    int marks = 80;  
  
    if (marks >= 50) {  
        cout << "You passed! ☑" << endl;  
    } else {  
        cout << "You failed! ✕" << endl;  
    }  
  
    return 0;  
}
```

🔍 Output:

You passed! ☑

🔄 Flow:

If (condition is true) → execute 'if'
Else → execute 'else'

◇ 3. if-else if Statement

☑ Analogy:

Checking who is knocking:

- If it's 👤 → open door
- Else if it's a delivery guy 📦 → take parcel
- Else → ignore

🔧 Code:

```
#include <iostream>  
using namespace std;  
  
int main() {  
    int speed = 80;  
  
    if (speed > 100) {  
        cout << "Too fast! 🚗" << endl;  
    } else if (speed > 60) {  
        cout << "Drive carefully 🚗" << endl;  
    }  
}
```

```
    return 0;
}
```

🔍 Output:

Drive carefully 🚗

📄 Flow:

Check 1st condition → true? → run block ☒ and skip rest
Check 2nd condition → only if 1st was false

◇ 4. if-else if-else Chain

☑ Analogy:

Exam results:

- If ≥ 90 → Grade A 🏆
- Else if ≥ 70 → Grade B 🥈
- Else → Try again! 🔄

🔧 Code:


```
#include <iostream>
using namespace std;

int main() {
    int score = 65;

    if (score >= 90) {
        cout << "Grade A 🏆" << endl;
    } else if (score >= 70) {
        cout << "Grade B 🥈" << endl;
    } else {
        cout << "Grade C 🔄" << endl;
    }


    return 0;
}
```

🔍 Output:

Grade C  Flow:

```
Check condition1 → true? Run & stop  
Else check condition2 → true? Run & stop  
Else → default block
```

◇ 5. Nested `if` Statements

 Analogy:

You ask: Do I have time? → If yes, then check: Do I have internet? → If yes → Watch Netflix 📺 → Else → Read book 📖 Else → Sleep 😴

 Code:

```
#include <iostream>  
using namespace std;  
  
int main() {  
    bool haveTime = true;  
    bool haveInternet = false;  
  
    if (haveTime) {  
        if (haveInternet) {  
            cout << "Watch Netflix 📺" << endl;  
        } else {  
            cout << "Read a book 📖" << endl;  
        }  
    } else {  
        cout << "Sleep 😴" << endl;  
    }  
  
    return 0;  
}
```




 Output:

Read a book 📖



 Flow:

```
Outer if → true?  
  ↓  
Inner if → true? Do A  
           → false? Do B  
Else → Do C
```


← Summary Flow Table

 Type	 Checks	 Decision Path
if	1	True = run block; False = do nothing
if-else	1	True = if block, False = else block
if-else if	2+	Checks in order until first true
if-else if-else	2+	Final else = fallback
Nested if	Multi	One inside another, used for combined decisions

Final Tip:

 Always use `{ }` braces for clarity, especially in nested or multi-line blocks  Add `cin.fail()` checks with `if-else` to catch bad inputs

Grade Evaluation using `if-else if`

	Real-world Analogy
You appeared for an exam. Based on your marks, the system gives you a grade:	
- 🏆 90 and above → Grade A	
- 🥈 70 to 89 → Grade B	
- 📖 50 to 69 → Grade C	
- ❌ Below 50 → Fail	

Code Example

```
#include <iostream>  
using namespace std;  
  
int main() {  
    int marks;  
    cout << "Enter your marks (0-100): ";  
    cin >> marks;  
  
    if (marks >= 90) {  
        cout << "Grade A 🏆" << endl;  
    }
```

```

    } else if (marks >= 70) {
        cout << "Grade B 🏆" << endl;
    } else if (marks >= 50) {
        cout << "Grade C 📖" << endl;
    } else {
        cout << "You Failed ❌" << endl;
    }

    return 0;
}

```



Output (Example for marks = 72)

Grade B 🏆



Flow Explanation

💡 The program checks conditions **from top to bottom**:

- `if (marks >= 90)` → ❌ false
- `else if (marks >= 70)` → ✅ true → executes this block
- **All further checks are skipped** once a match is found



If no condition matches, the `else` block runs as the **fallback/default**



Grade Evaluation using `if-else if-else`



Real-world Analogy

You got your exam result. Based on your marks:

- 🏆 If `marks >= 90` → Grade A
- 🏆 Else if `marks >= 70` → Grade B
- 📖 Else if `marks >= 50` → Grade C
- ❌ Else (below 50) → Fail



Code Example

```

#include <iostream>
using namespace std;

int main() {
    int marks;
    cout << "Enter your marks (0-100): ";
    cin >> marks;

    if (marks >= 90) {
        cout << "Grade A 🏆" << endl;
    }
}

```

```
    } else if (marks >= 70) {  
        cout << "Grade B 🏆" << endl;  
    } else if (marks >= 50) {  
        cout << "Grade C 📖" << endl;  
    } else {  
        cout << "You Failed ❌" << endl;  
    }  
  
    return 0;  
}
```



Sample Outputs

Input: 92 → Output: Grade A 🏆
Input: 78 → Output: Grade B 🏆
Input: 55 → Output: Grade C 📖
Input: 43 → Output: You Failed ❌



Flow Explanation

► **Condition 1:** marks >= 90

☑ If true → Grade A and exit

❌ Else go to next

► **Condition 2:** marks >= 70

☑ If true → Grade B and exit

► **Condition 3:** marks >= 50

☑ If true → Grade C and exit

► **Else** (fallback): if none match → You Failed ❌

☑ Only **one block** executes — the **first one** that passes