UNIVERSITAS
GADJAH MADA

Meeting 4

Program control: Selection & Iteration

Fundamentals of Programming
TKU211131

DTETI FT UGM

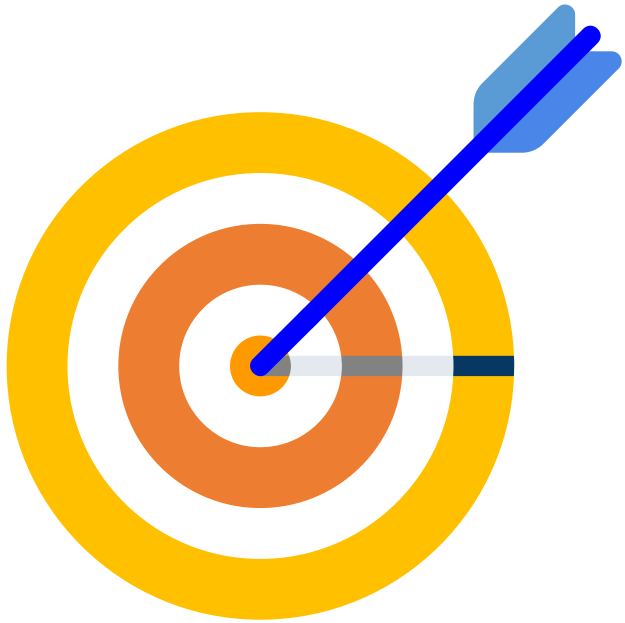
Bimo Sunarfri Hantono
bhe@ugm.ac.id

LOCALLY ROOTED,
GLOBALLY RESPECTED

ugm.ac.id

Topics

- Understand control structures in programming.
- Explain the role of **selection and iteration**.
- Implement:
  - if, if-else, nested if, switch-case (**selection**)
  - for, while, do–while (**iteration**)
- Apply **selection and iteration** in biomedical, electrical, and IT contexts.

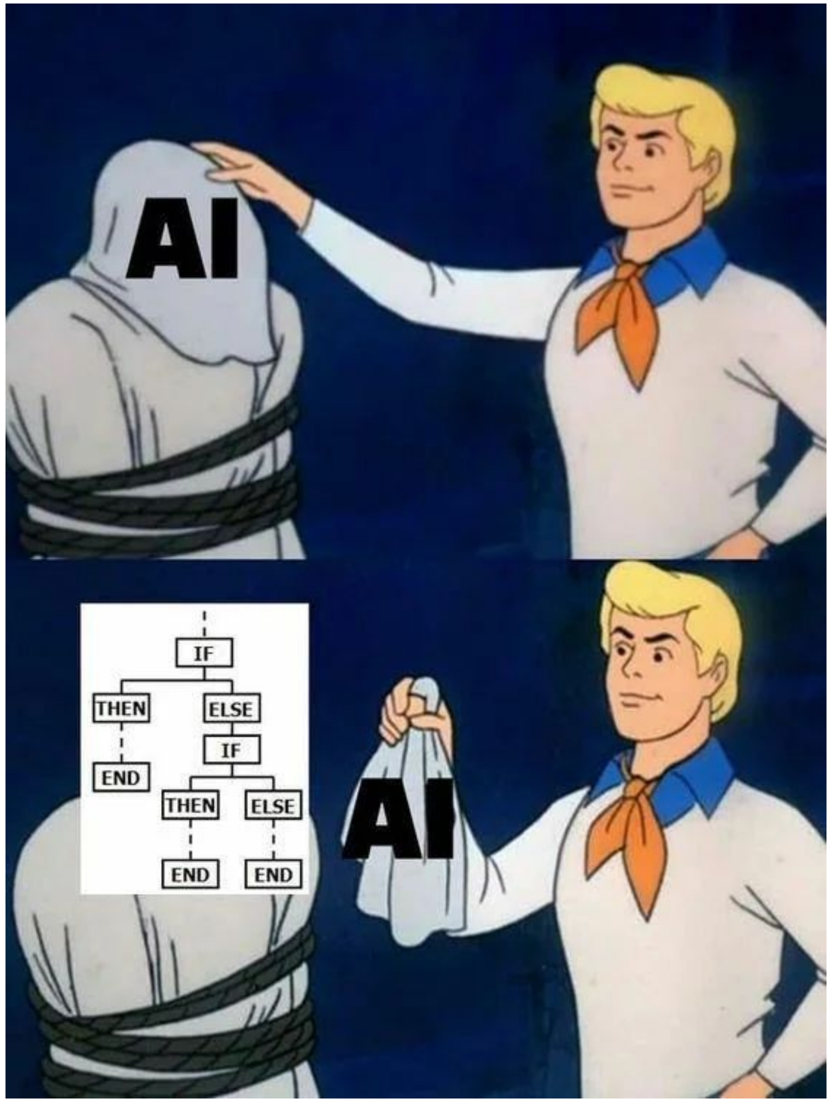


UNIVERSITAS
GADJAH MADA

Program Controls

Why Program Control?

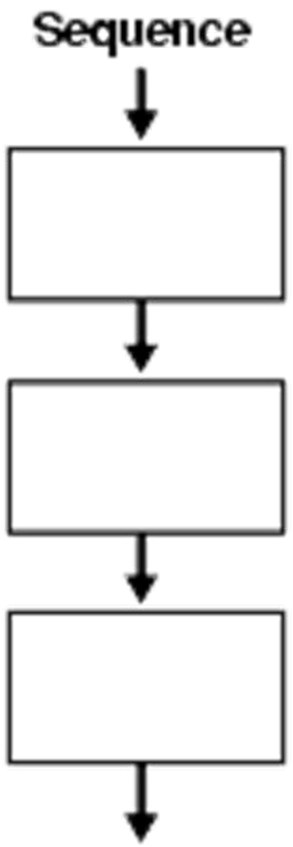
- Programs rarely run straight from top to bottom
- Real-world problems require decisions
- Selection = choosing the right path

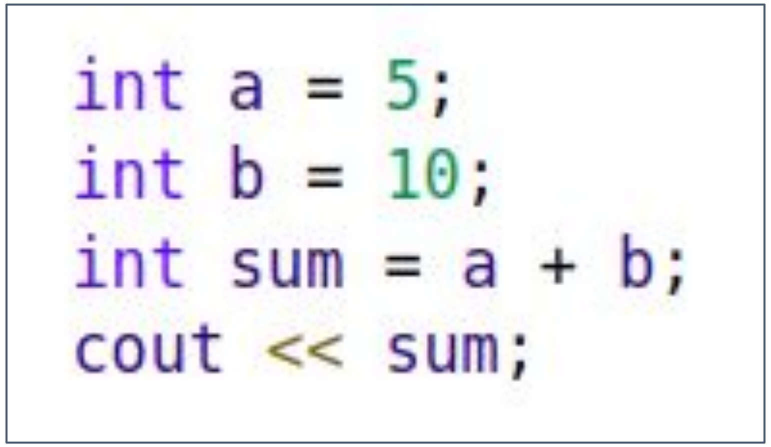


Types of Program Controls (1)

There are 4 types of program controls:

- **Sequence**
  - Instructions executed in order, step by step.
  - No decision making.



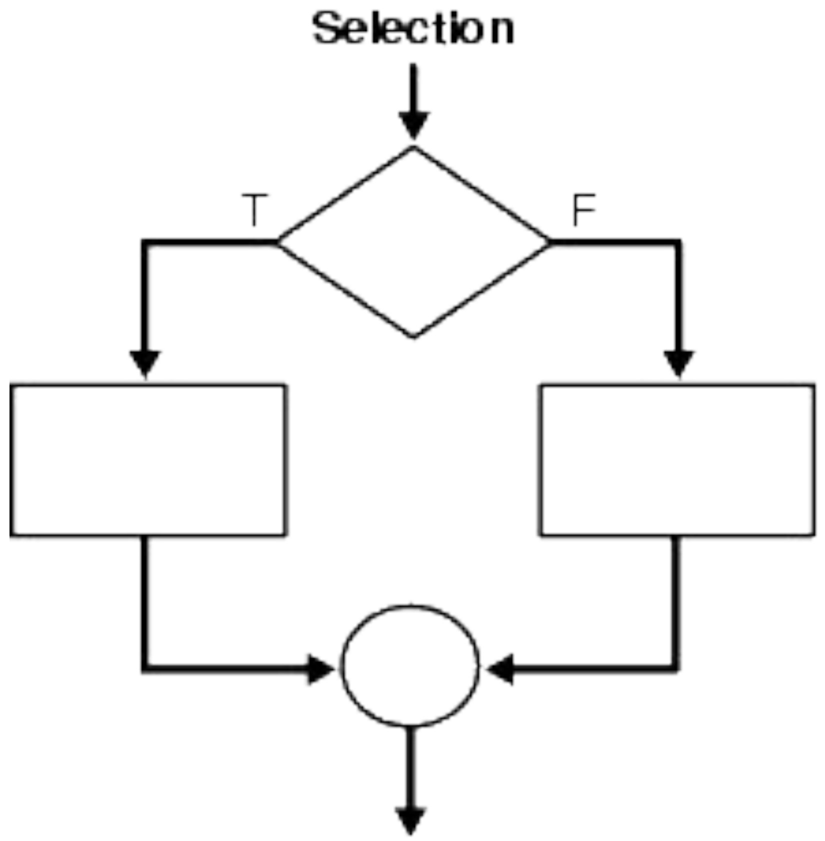


```
int a = 5;
int b = 10;
int sum = a + b;
cout << sum;
```

Types of Program Controls (2)

There are 4 types of program controls:

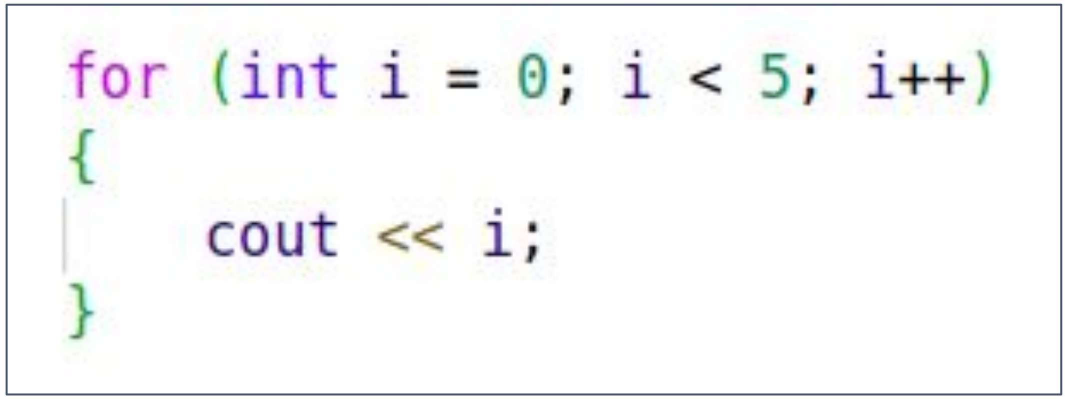
- **Selection**
  - Decision making based on conditions.
  - Uses Boolean logic (true/false).
  - Examples: if, if-else, switch.



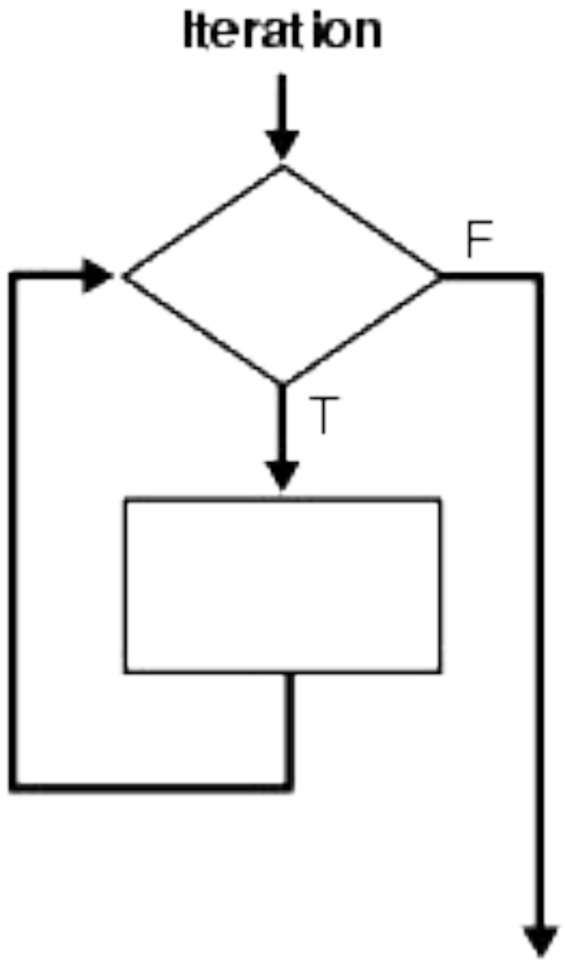
Types of Program Controls (3)

There are 4 types of program controls:

- **Iteration or loop or Repetition**
  - Repeat until condition met.
  - Types: for, while, do-while.
  - Example:



```
for (int i = 0; i < 5; i++)
{
    cout << i;
}
```

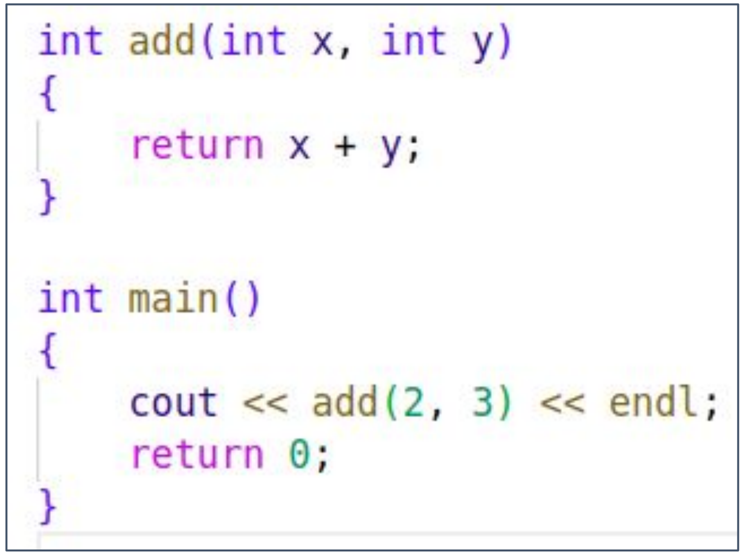


Types of Program Controls (4)

UNIVERSITAS GADJAH MADA

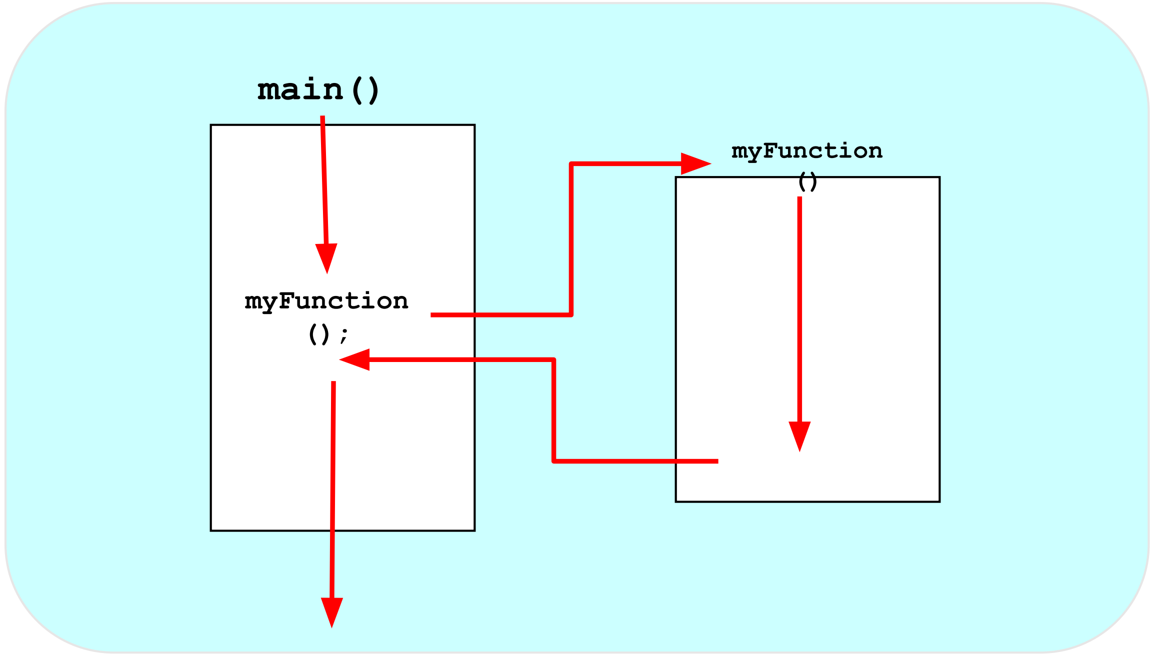
There are 4 types of program controls:

- **Subprogram**
  - Break program into reusable parts.
  - Functions & procedures.



```
int add(int x, int y)
{
    return x + y;
}

int main()
{
    cout << add(2, 3) << endl;
    return 0;
}
```

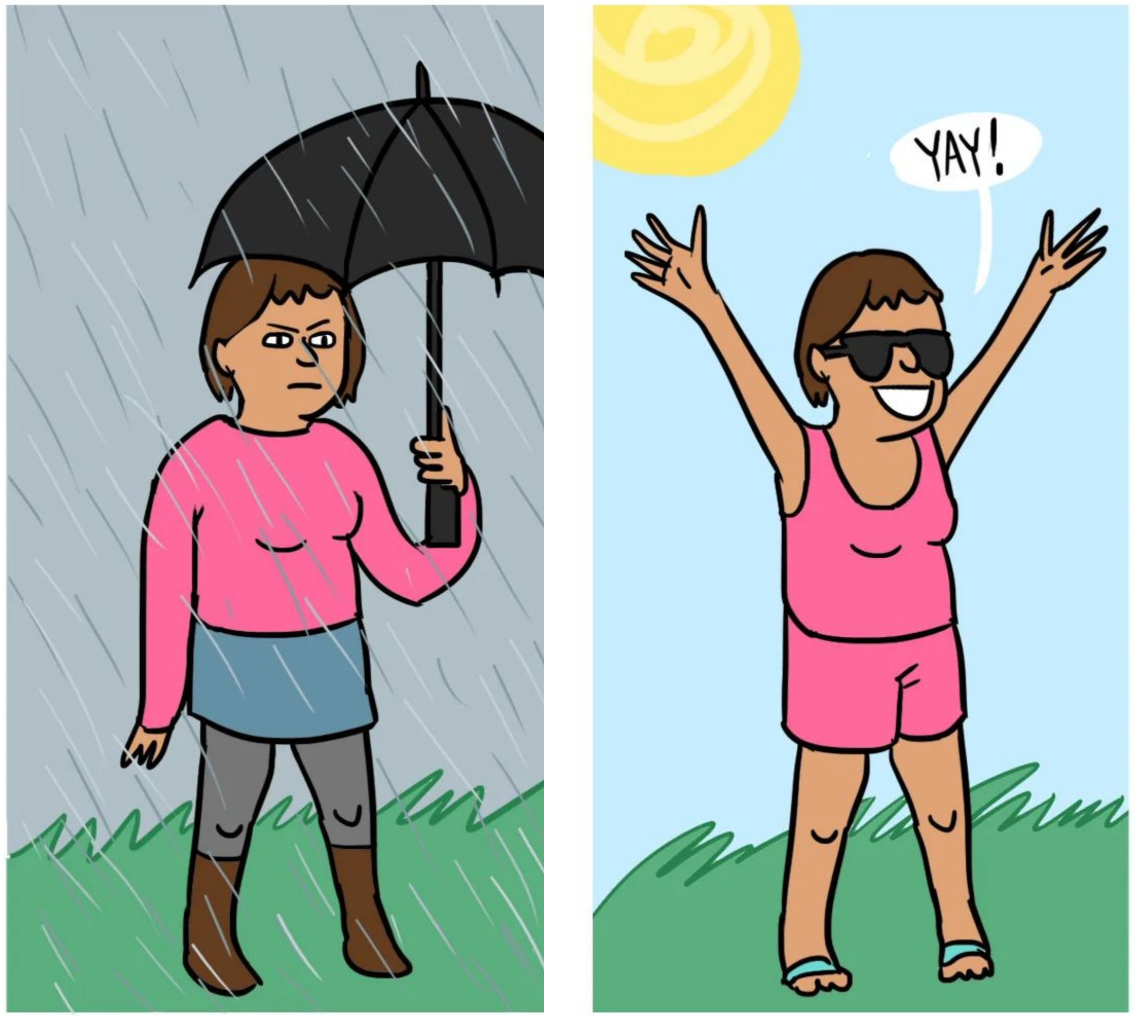
Subprogram



Real Life Analogy

UNIVERSITAS GADJAH MADA

- If it rains → bring umbrella
- Else → wear sunglasses



UNIVERSITAS GADJAH MADA

Program Controls: Selection

Boolean Expressions

- Conditions often use one of C++'s equality operators or relational operators, all of which produce boolean results:

  == equal to

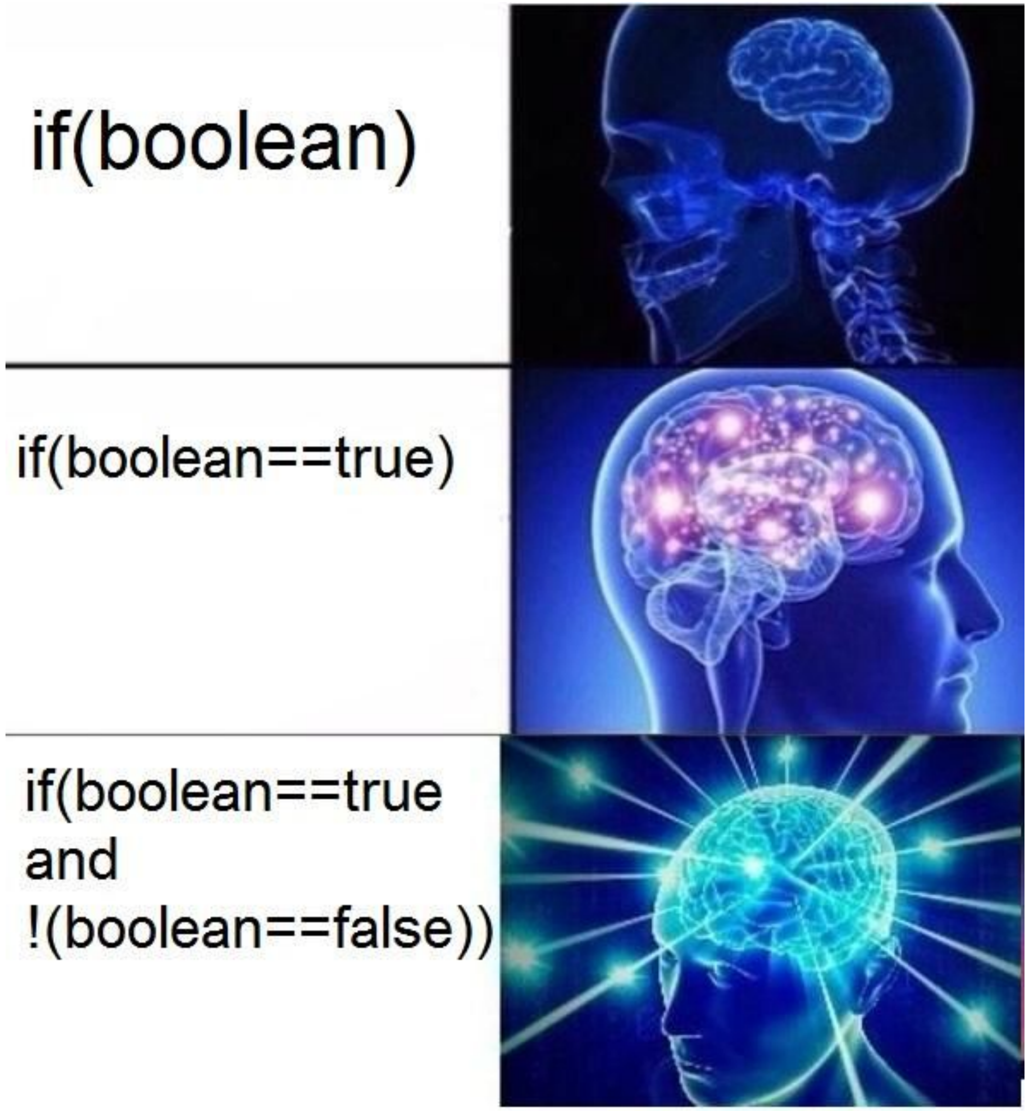
  != is not equal to

  < less than

  > more than

  <= less than or equal to

  >= more than or equal to

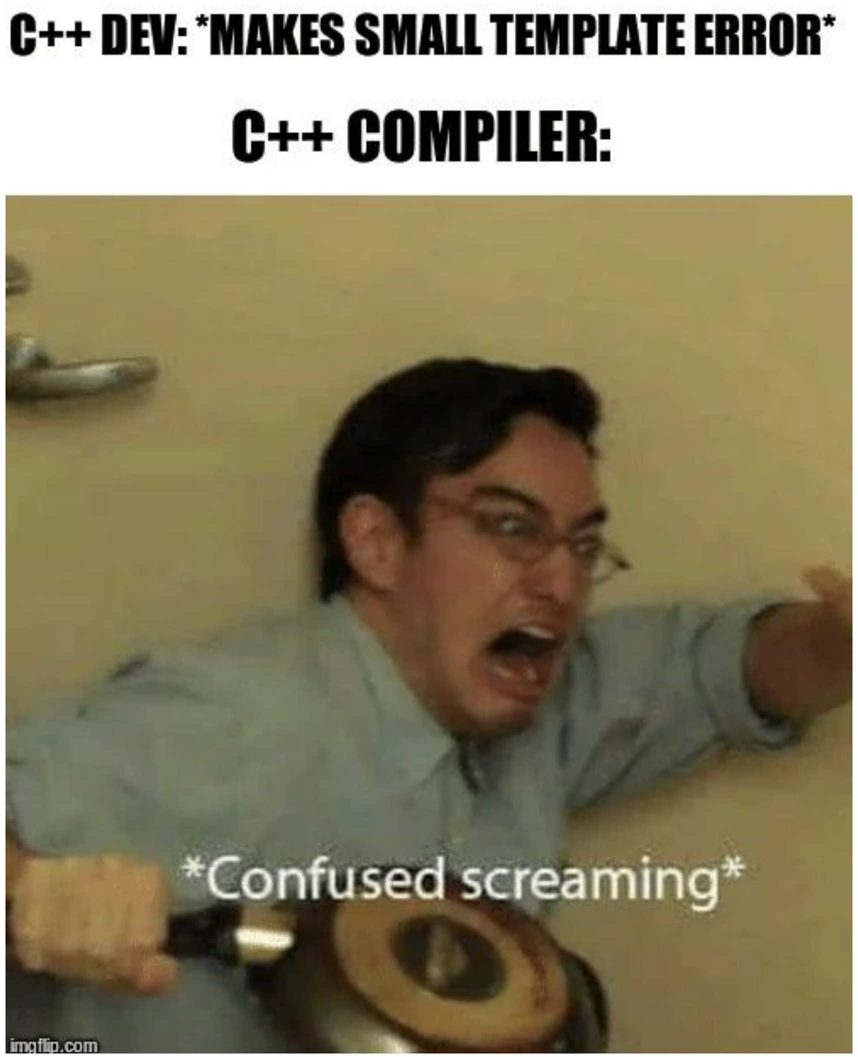


Common Pitfall

UNIVERSITAS GADJAH MADA

- Note the difference between the equality operator (==) and the assignment operator (=).

- Wrong: if (x = 5)
- Correct: if (x == 5)



UNIVERSITAS
GADJAH MADA

Single Selection (IF statement)

IF Statement

UNIVERSITAS GADJAH MADA

- Syntax of If statement

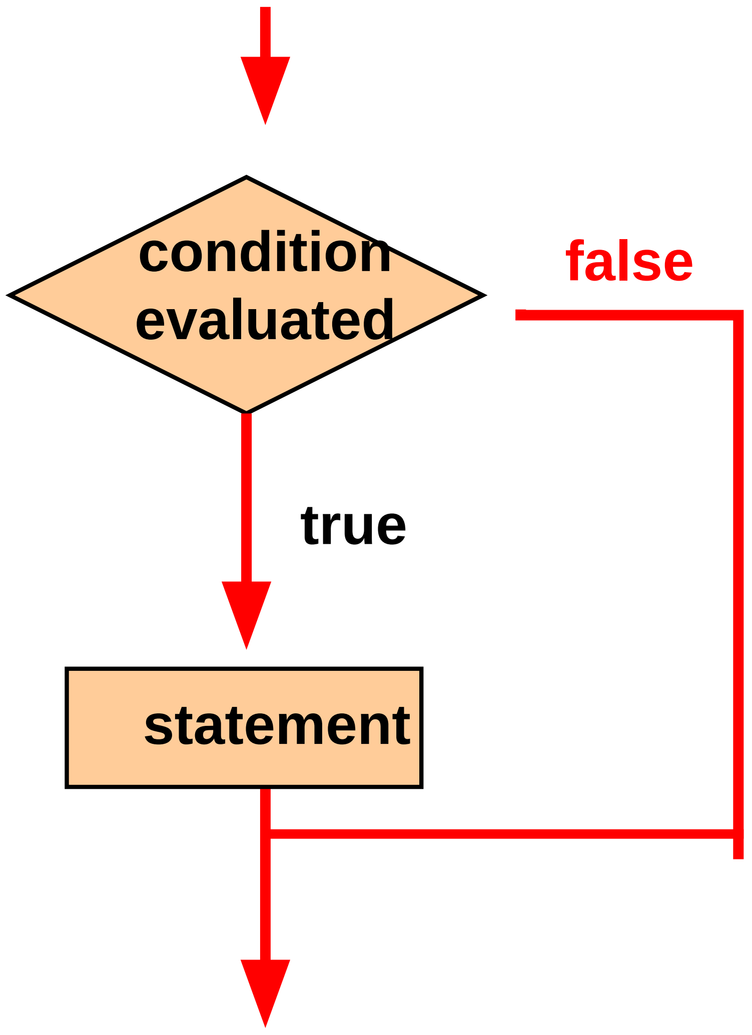
if is C++ reserved word

The *condition* must be a boolean expression. It must evaluate to either true or false.

```
if ( condition )
      statement;
```

If the *condition* is true, the ***statement*** is executed.
If it is false, the ***statement*** is skipped.

Logic of an IF Statement



UNIVERSITAS
GADJAH MADA

Double Selection (IF-ELSE statement)

IF-ELSE Statement

UNIVERSITAS GADJAH MADA

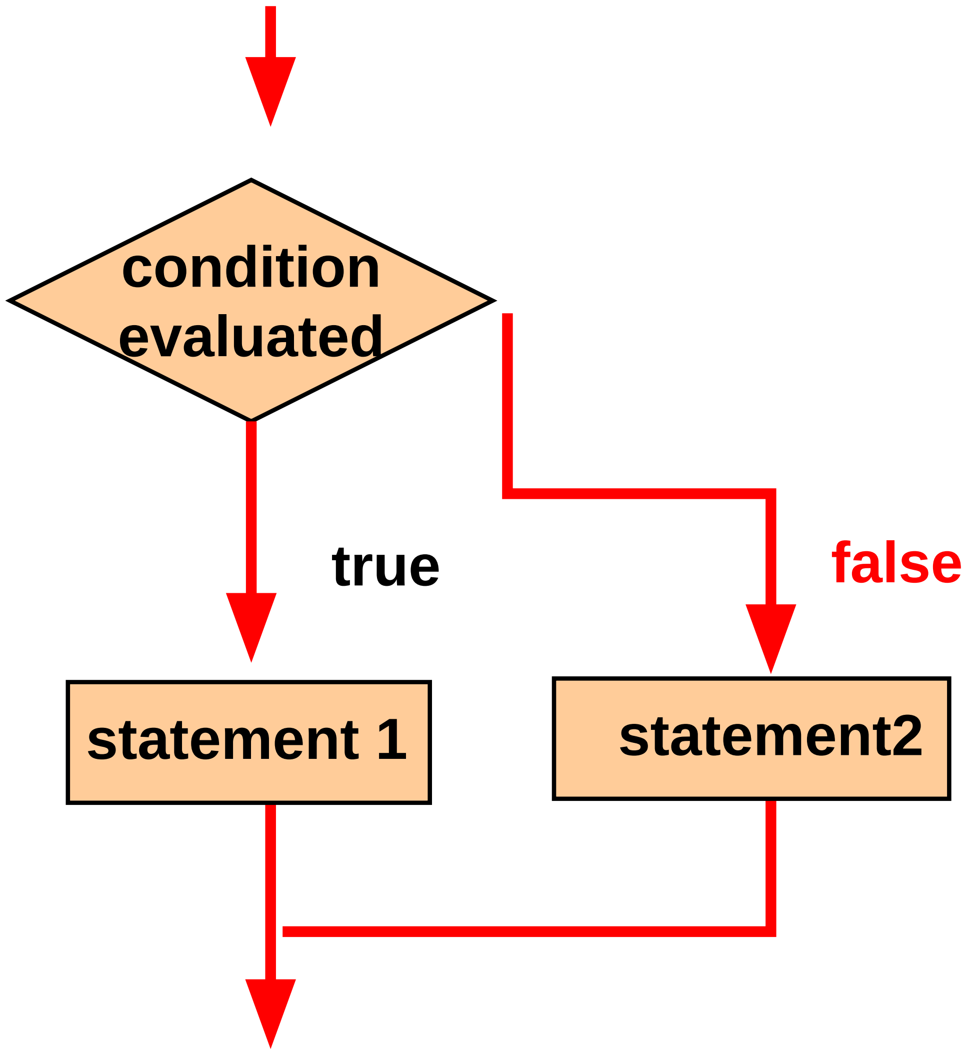
- An else clause can be added to an if statement to create an if-else statement.

```
if(condition)
     statement1;
     else
       statement2;
```

- If the condition is true, statement1 is executed;
- If the condition is false, statement2 is executed.
- One of the two will be executed, but not both.

Logic of an IF-ELSE Statement

UNIVERSITAS GADJAH MADA



UNIVERSITAS
GADJAH MADA

Multiple Selection (nested IF-ELSE statement)

Nested IF

UNIVERSITAS GADJAH MADA

- Nested if tests multiple cases by placing an if/else structure inside another if/else structure.
- If-else constructs can be placed one inside the other with any depth.

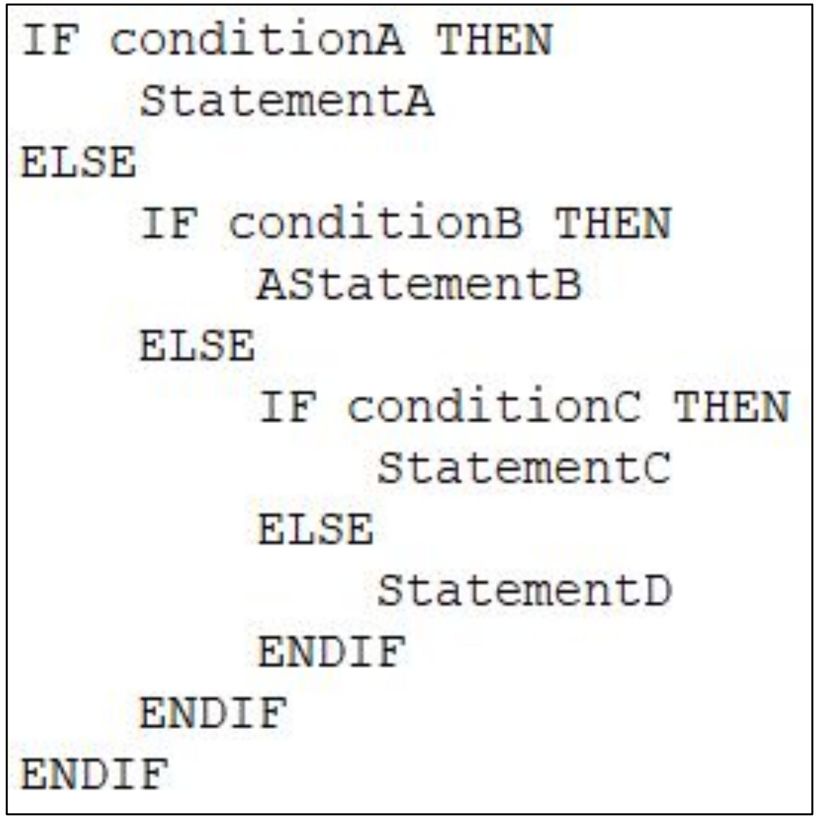
```
IF conditionA THEN
    StatementA
ELSE
    IF conditionA THEN
        StatementB
    ELSE
        IF conditionC THEN
            StatementC
        ELSE
            StatementD
        ENDIF
    ENDIF
ENDIF
```

Linear nested selection

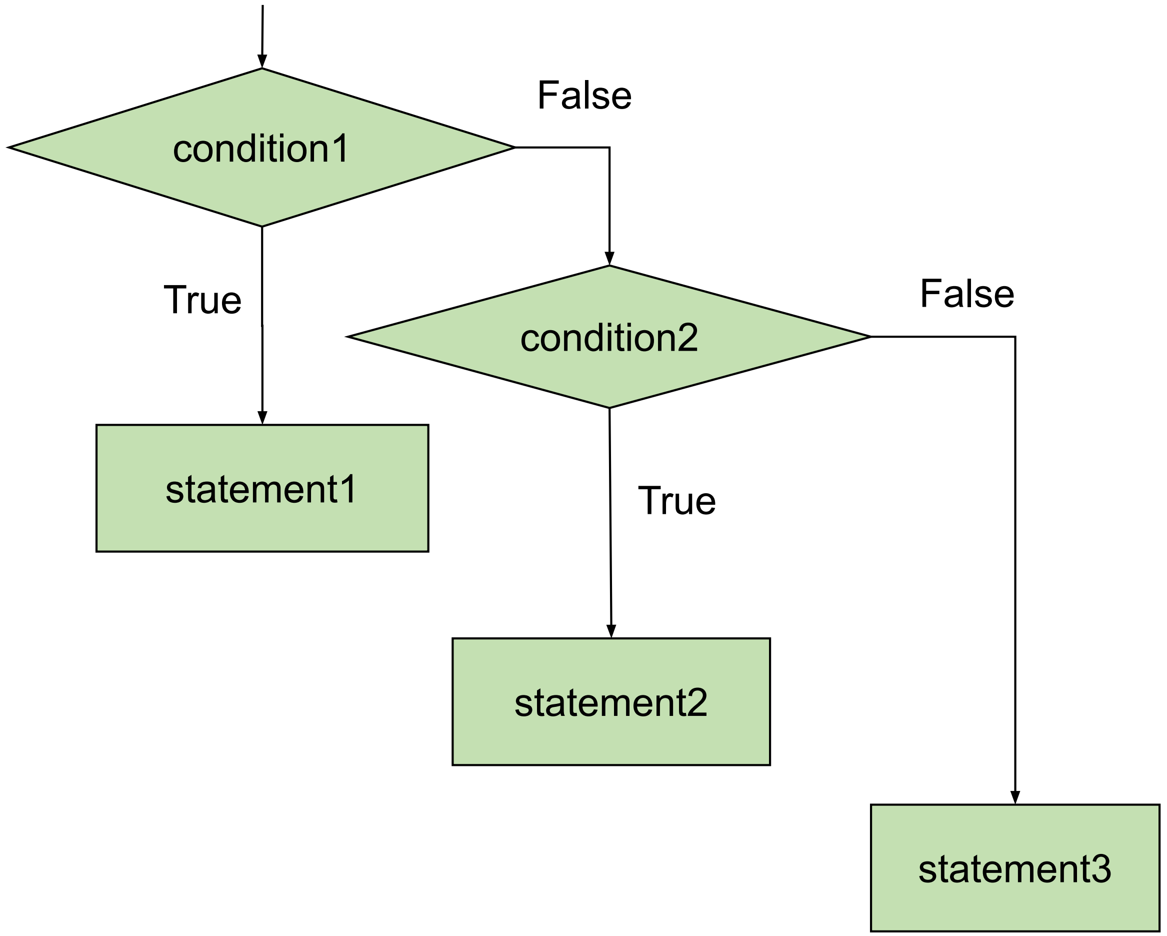
```
IF conditionA THEN
    IF conditionA1 THEN
        IF conditionA11 THEN
            StatementA11
        ELSE
            StatementA12
        ENDIF
    ELSE
        StatementA2
    ENDIF
ELSE
    StatementB
ENDIF
```

Non-linear nested selection

Pseudocode & Flowchart Linear Nested Selection

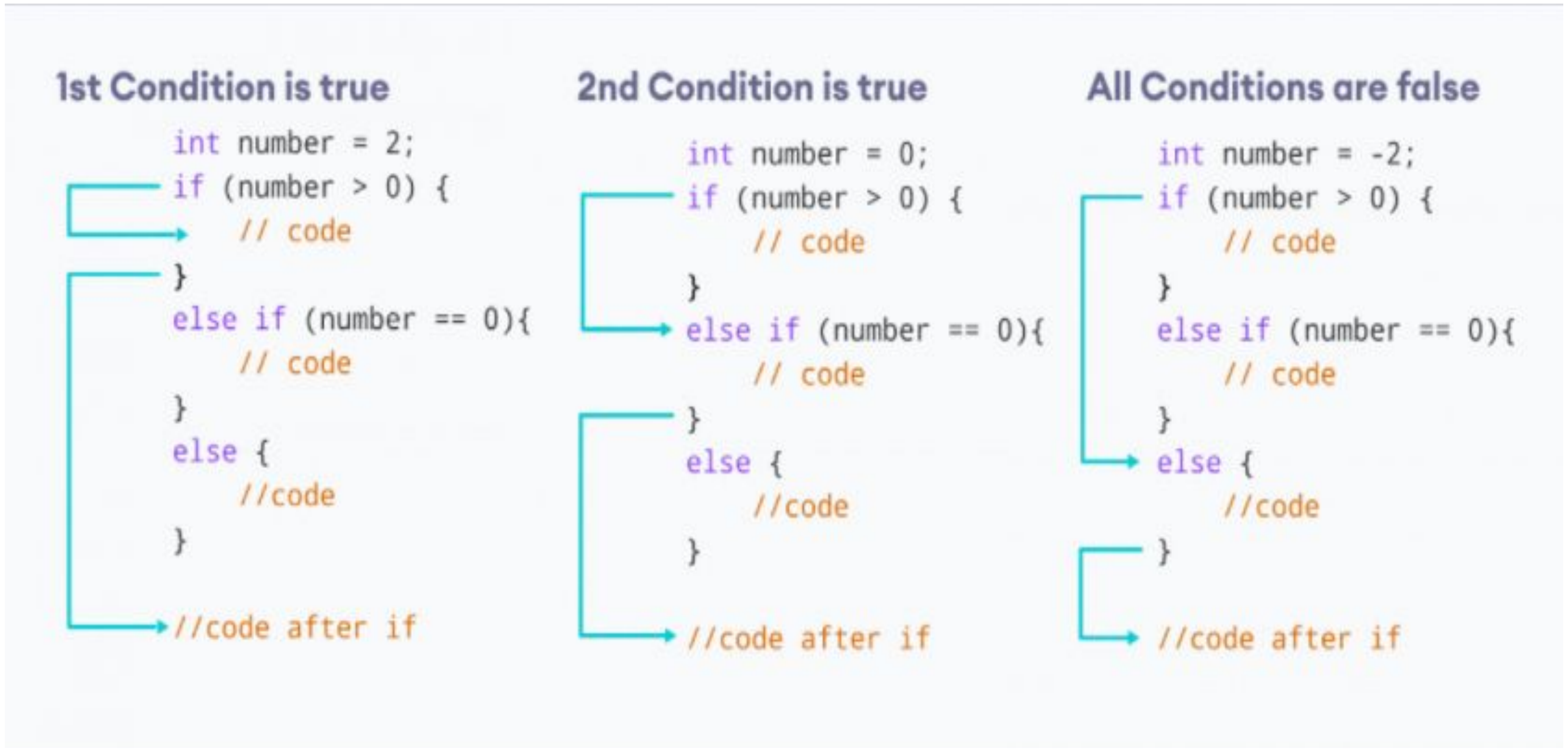


```
IF conditionA THEN
    StatementA
ELSE
    IF conditionB THEN
        AStatementB
    ELSE
        IF conditionC THEN
            StatementC
        ELSE
            StatementD
        ENDIF
    ENDIF
ENDIF
```



How Linear Nested Selection Work

UNIVERSITAS GADJAH MADA



1st Condition is true

```
int number = 2;
if (number > 0) {
    // code
}
else if (number == 0){
    // code
}
else {
    //code
}

//code after if
```

2nd Condition is true

```
int number = 0;
if (number > 0) {
    // code
}
else if (number == 0){
    // code
}
else {
    //code
}

//code after if
```

All Conditions are false

```
int number = -2;
if (number > 0) {
    // code
}
else if (number == 0){
    // code
}
else {
    //code
}

//code after if
```

Pseudocode Non-Linear Nested Selection

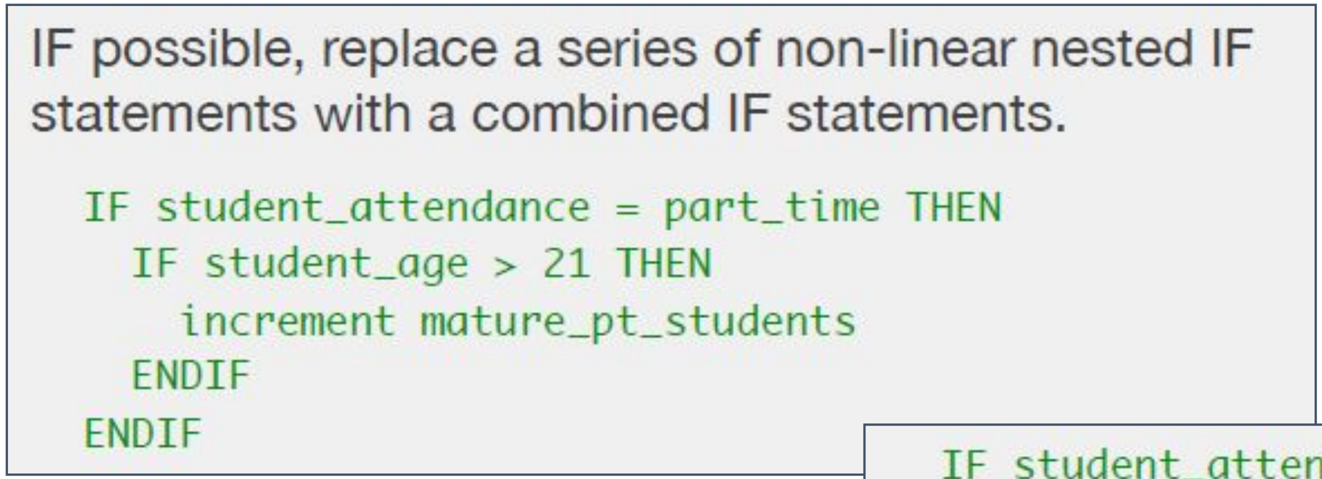
```
if(condition_1)
   if(condition_2)
      if(condition_3)
         statement_4;
      else
         statement_3;
   else
      statement_2;
else
   statement_1;
next_statement;
```

Linear vs Non-Linear Nested Selection

- **Linear Nested Selection**
  - The if–else if–else structure executes sequentially (linear).
  - Only **one condition** will be executed.
  - Used when conditions are **mutually exclusive**.
  - **Example**: Classifying student grades (A, B, C, D).
- **Non-Linear Nested Selection**
  - An **if** statement can be placed inside another **if**.
  - Can check **combinations of conditions**.
  - Used when a decision depends on **multiple layered conditions**.
  - **Example**: Login validation (check username first → then check password).

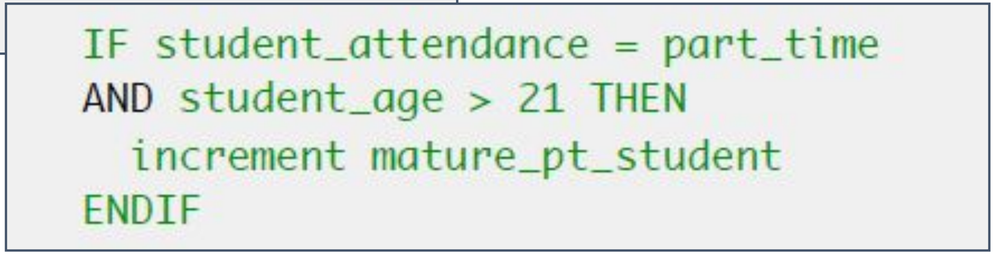
Combined IF

- Use operators like AND (&&) and OR (||)
  - AND: both conditions must be TRUE
  - OR: minimum one condition must be TRUE



IF possible, replace a series of non-linear nested IF statements with a combined IF statements.

```
IF student_attendance = part_time THEN
  IF student_age > 21 THEN
    increment mature_pt_students
  ENDIF
ENDIF
```



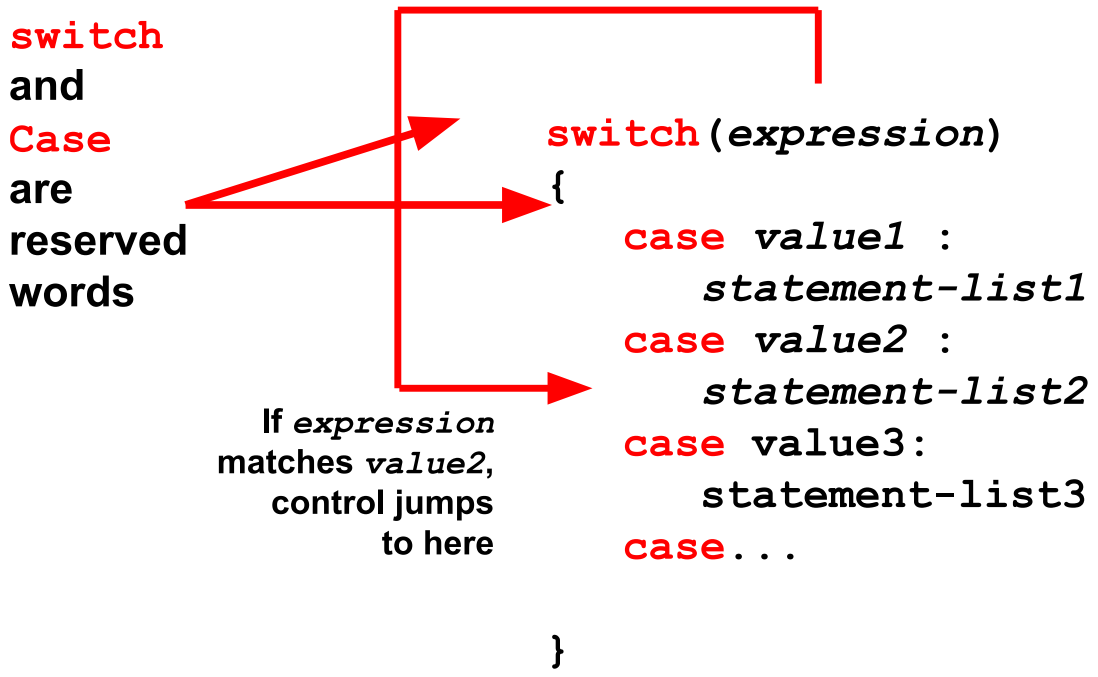
```
IF student_attendance = part_time
AND student_age > 21 THEN
  increment mature_pt_student
ENDIF
```

UNIVERSITAS GADJAH MADA

Multiple Selection (SWITCH-CASE-BREAK statement)

Switch Case

- Switch = flexible selection control.
- Handles **multiple choices** (not just true/false).
- Alternative to many if–else if.
- Works with **integral types**: int, char.


switch and Case are reserved words
If *expression* matches *value2*, control jumps to here

```
switch(expression)
{
    case value1 :
        statement-list1
    case value2 :
        statement-list2
    case value3:
        statement-list3
    case...

}
```

Behavior of Break & Default

UNIVERSITAS GADJAH MADA

- **Break**
  a. Ends case execution.
  b. Without it → control flows to next case (fall-through).

- **Default**
  a. Executes when no other case matches.
  b. Acts like “else” in if–else.

Flowchart Switch Case

UNIVERSITAS GADJAH MADA

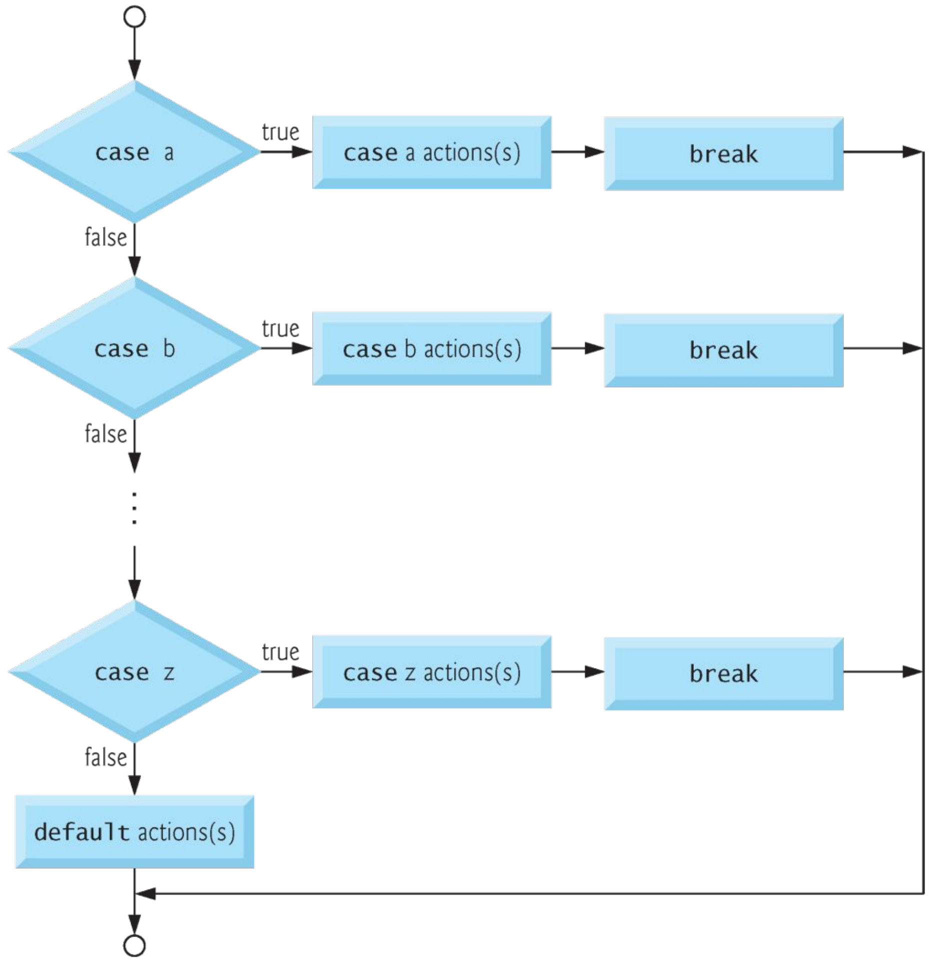


Fig. 4.8 | switch multiple-selection statement with breaks.

```
switch(condition)
{
    case template_1 : statement(s);
                break;
    case template_2 : statement(s);
                break;
    case template_3 : statement(s);
                break;
    ...
    ...
    case template_n : statement(s);
                break;

    default: statement(s);
}
next_statement;
```

Switch vs Nested If: When to Use Which?

UNIVERSITAS GADJAH MADA

Aspect	`switch`	`if / else if` **(nested)**
Input type	Discrete, integral (`int`, `char`)	Any boolean expression (ranges, relations, combos)
Number of branches	Many discrete options	Few to many; good for **ranges**
Readability	Very clean for enums/menus/modes	Clear for comparisons like `<, >, <=, >=`
Fall-through	Possible without `break`	Not applicable
Default/else	`default` for "other cases"	`else` for "otherwise"
Performance	slightly faster for many discrete cases	Comparable; depends on optimizer

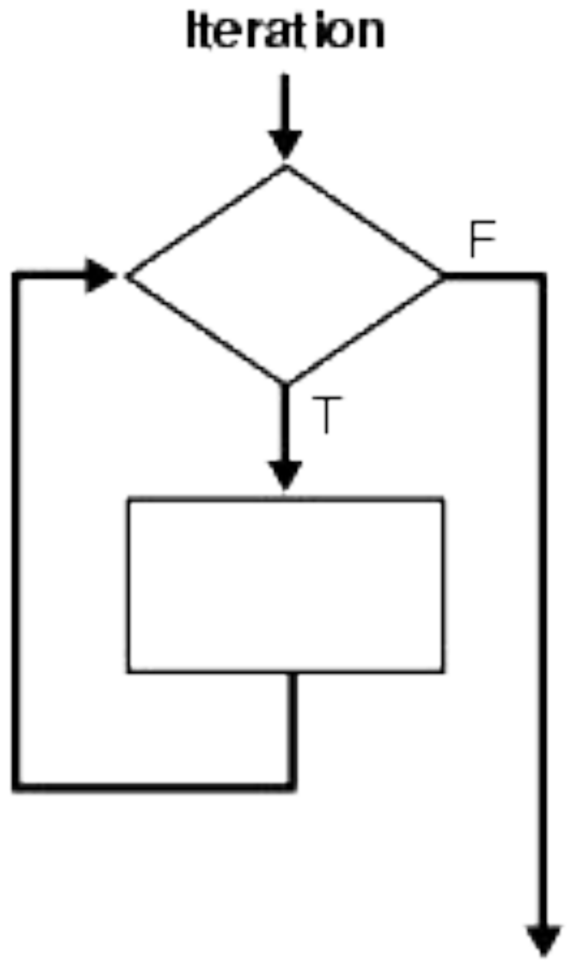
- Use switch when: value is discrete and mutually exclusive (menu options, protocol codes, device modes, enum states).
- Use nested if when: you need relational checks or ranges (e.g., temp > 37.5 && spo2 < 92), or complex boolean logic.

UNIVERSITAS GADJAH MADA

Program Controls: Iteration

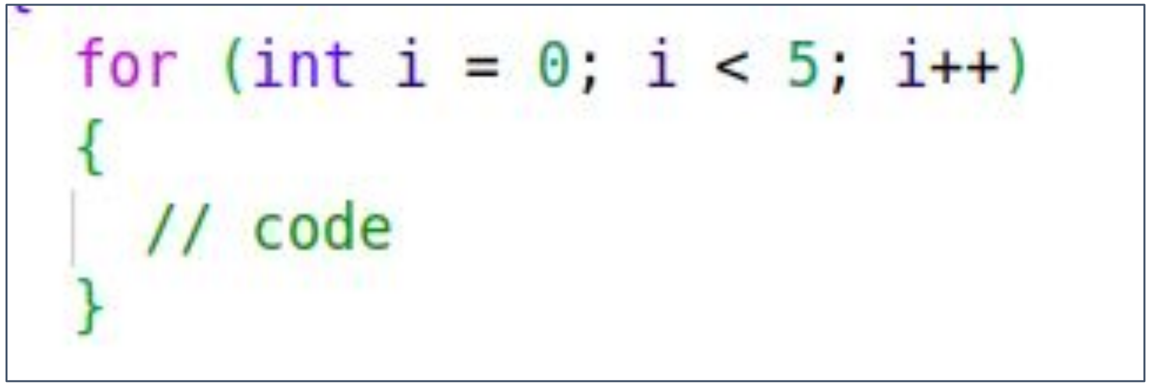
Intro to Iteration

- Iteration = repeating a block of code multiple times.
- Saves time, avoids redundancy.
- Types of loops in C++:
  1. **for loop**
  2. **while loop**
  3. **do–while loop**



For Loop

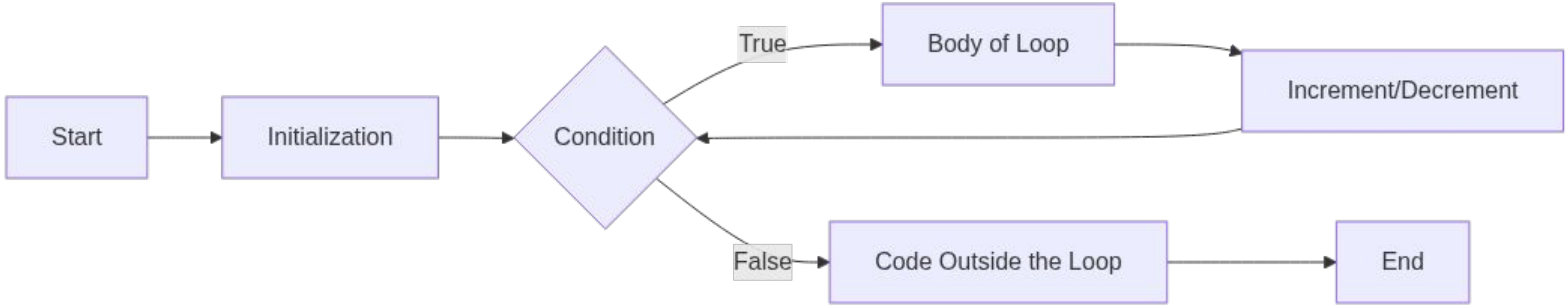
- Best when the number of repetitions is **known in advance**.
- Common uses:
    - Counting loops (1 to N).
    - Iterating through arrays or collections.
    - Generating tables or patterns.
- Combines **initialization, condition, update** in one line.



```
for (int i = 0; i < 5; i++)
{
   // code
}
```

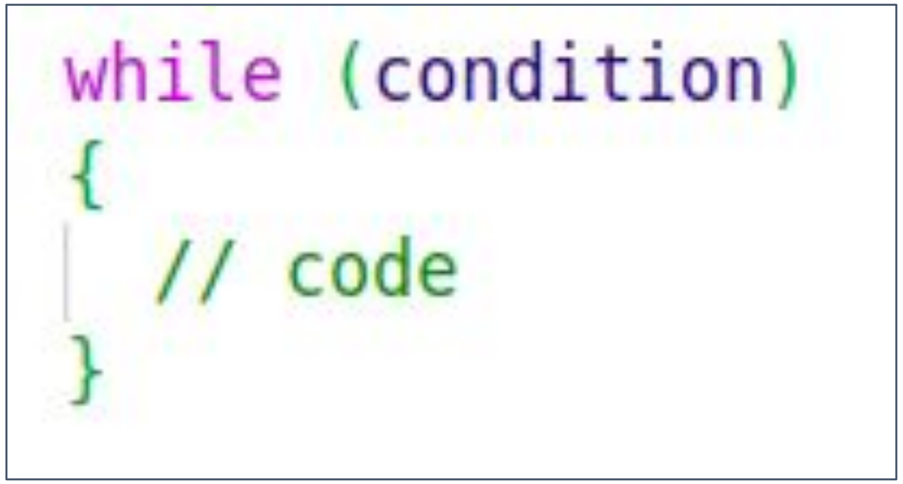
Logic of an For Loop

UNIVERSITAS GADJAH MADA



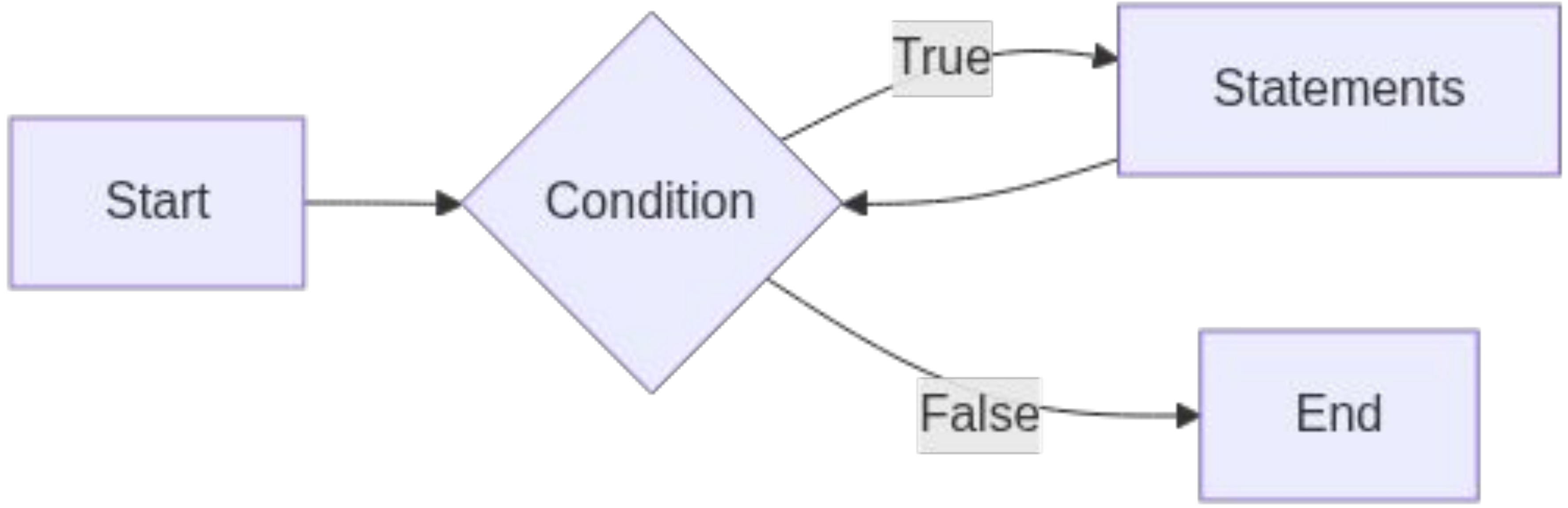
While Loop

- Best when the number of repetitions is **unknown**.
- Runs **while condition is true**.
- Common uses:
  - Waiting for sensor/data input.
  - Monitoring a process until it ends.
  - User input validation.
- Risk: infinite loop if condition never becomes false.



```
while (condition)
{
    // code
}
```

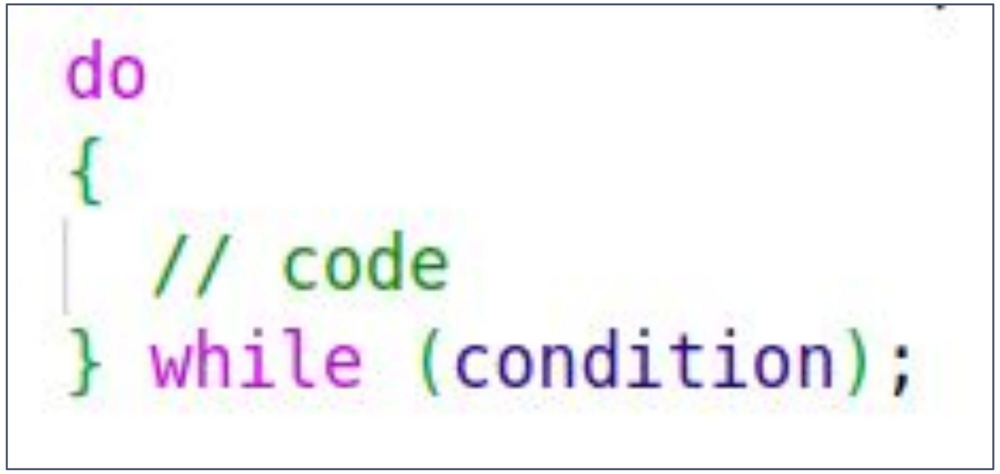
Logic of an While Loop



Do-While Loop

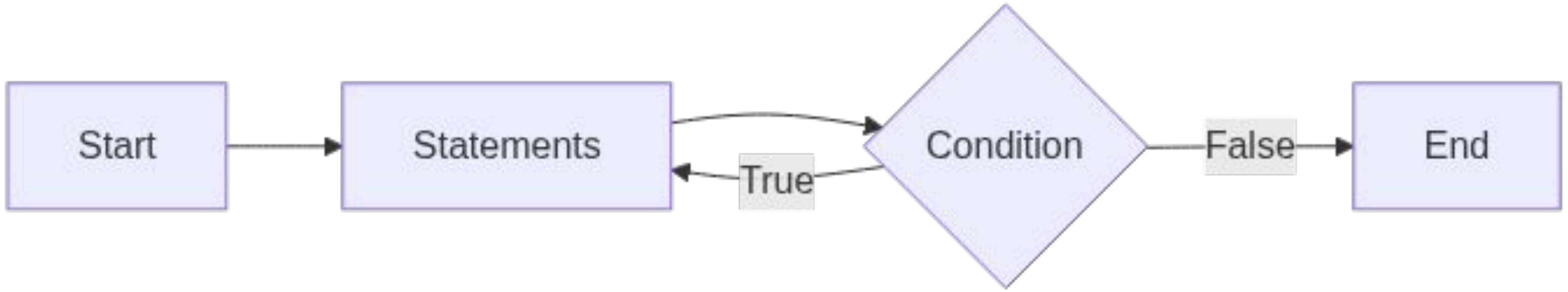
UNIVERSITAS GADJAH MADA

- Executes body **at least once**, condition checked at the end.
- Common uses:
  - Menu-driven programs (run at least once).
  - Retry until valid input is given.
  - Simulations where one iteration must always occur.
- Difference with while: **do–while always runs once**, while may not run at all.



```
do
{
  // code
} while (condition);
```

Logic of an Do-While Loop



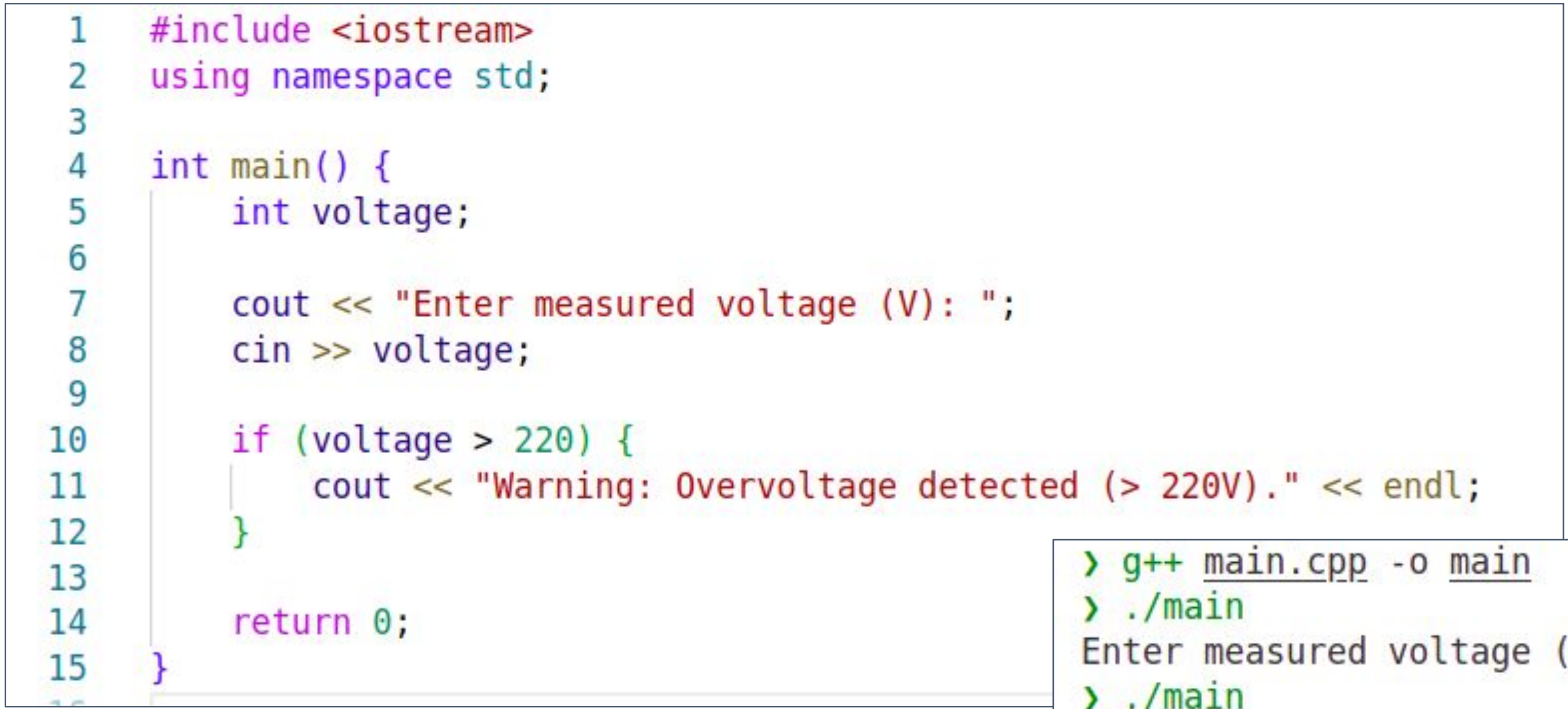
Any Question?

UNIVERSITAS GADJAH MADA

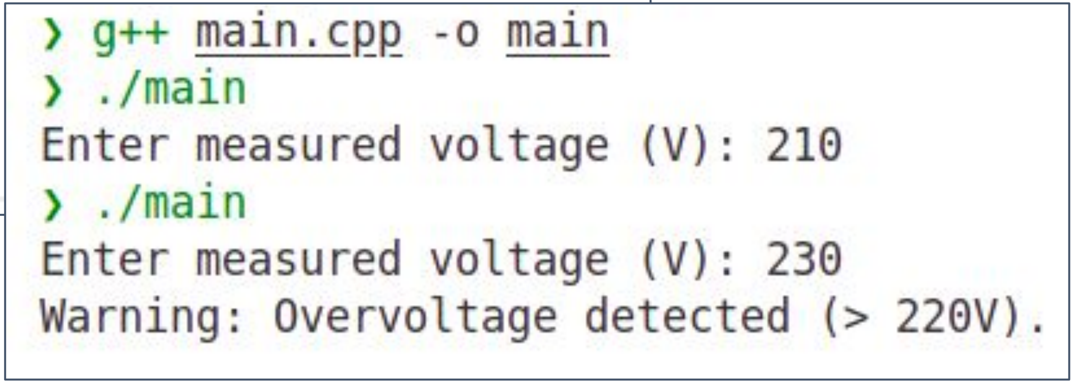
Example Code

Example of an IF Statement: Voltage Safety Check

UNIVERSITAS GADJAH MADA



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

int main() {
    int voltage;

    cout << "Enter measured voltage (V): ";
    cin >> voltage;

    if (voltage > 220) {
        cout << "Warning: Overvoltage detected (> 220V)." << endl;
    }

    return 0;
}
```

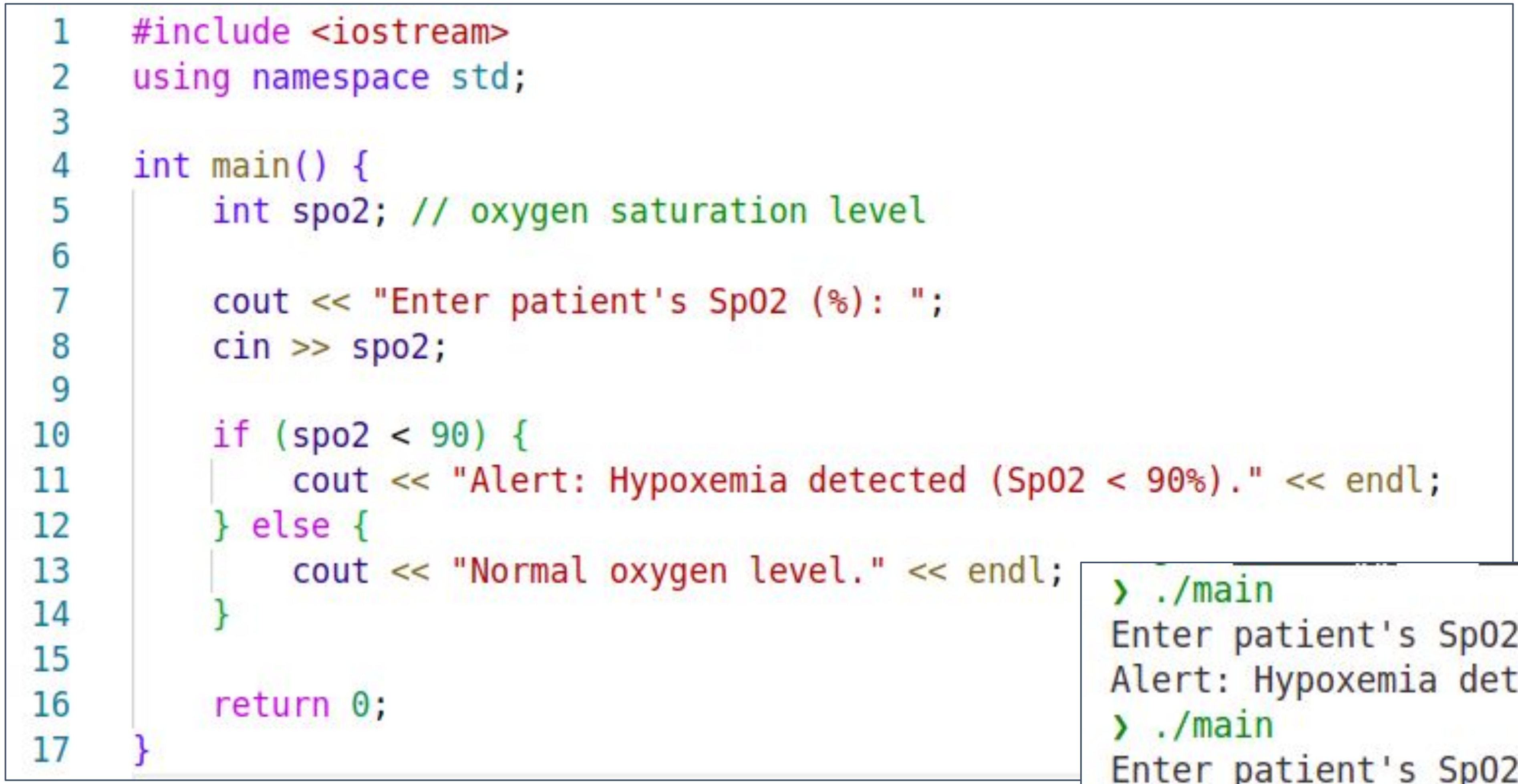


```
❯ g++ main.cpp -o main
❯ ./main
Enter measured voltage (V): 210
❯ ./main
Enter measured voltage (V): 230
Warning: Overvoltage detected (> 220V).
```

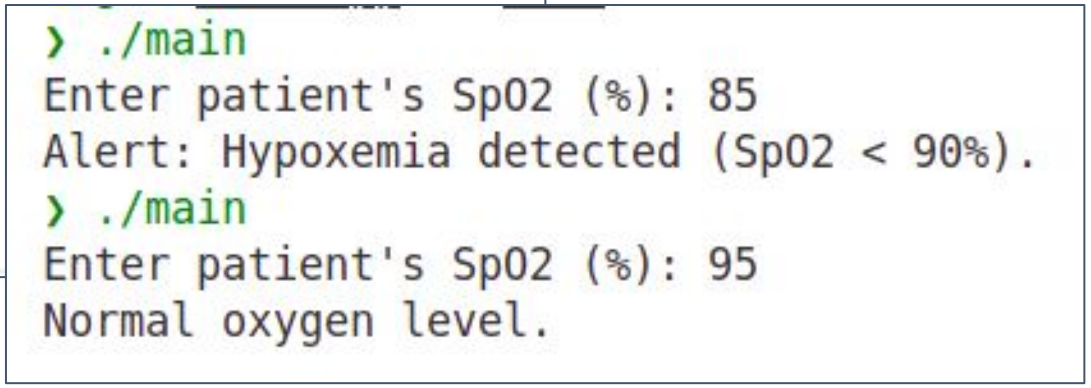
Example of an IF-ELSE Statement:

UNIVERSITAS GADJAH MADA

Oxygen Saturation Check

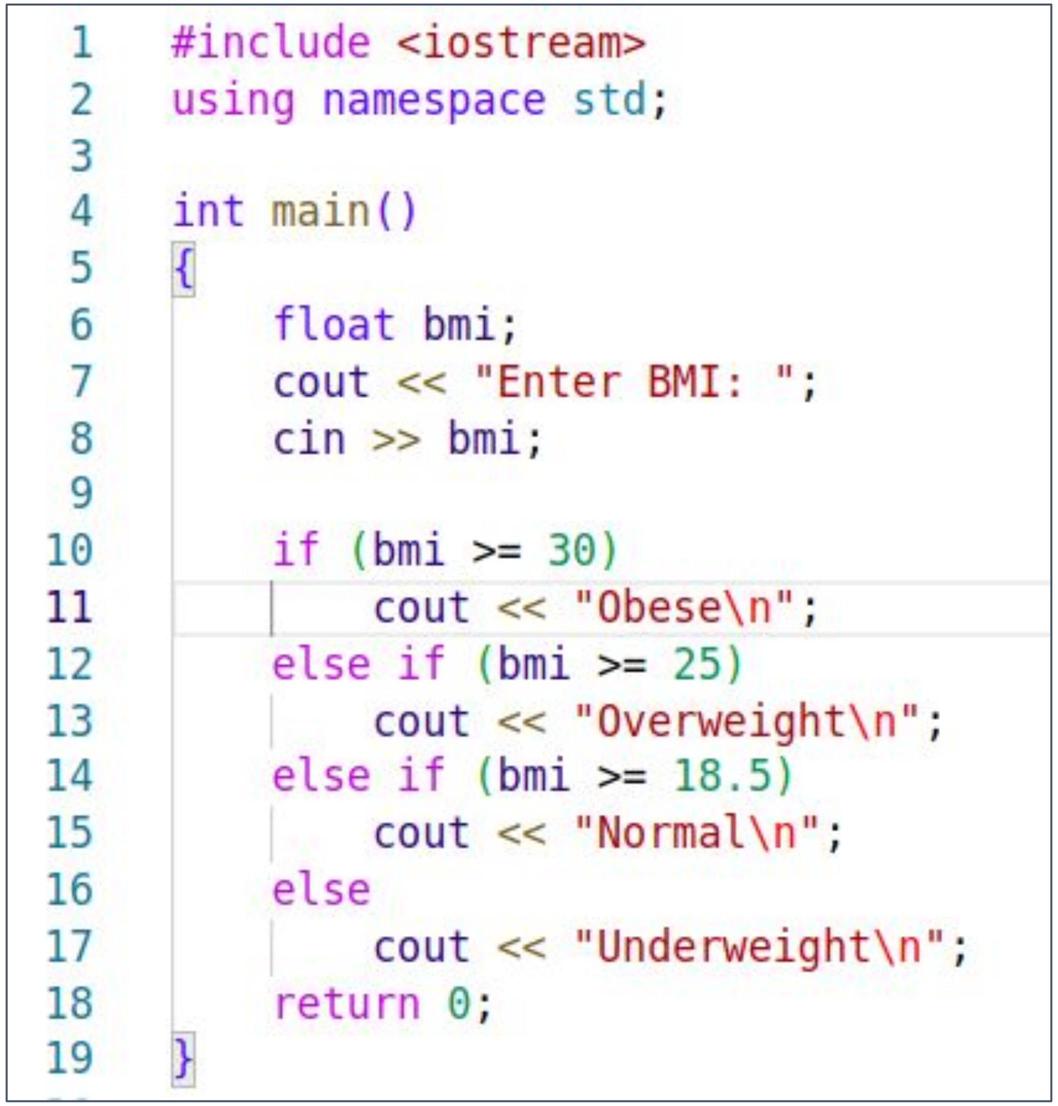


```cpp
#include <iostream>
using namespace std;

int main() {
    int spo2; // oxygen saturation level

    cout << "Enter patient's SpO2 (%): ";
    cin >> spo2;

    if (spo2 < 90) {
        cout << "Alert: Hypoxemia detected (SpO2 < 90%)." << endl;
    } else {
        cout << "Normal oxygen level." << endl;
    }

    return 0;
}
```



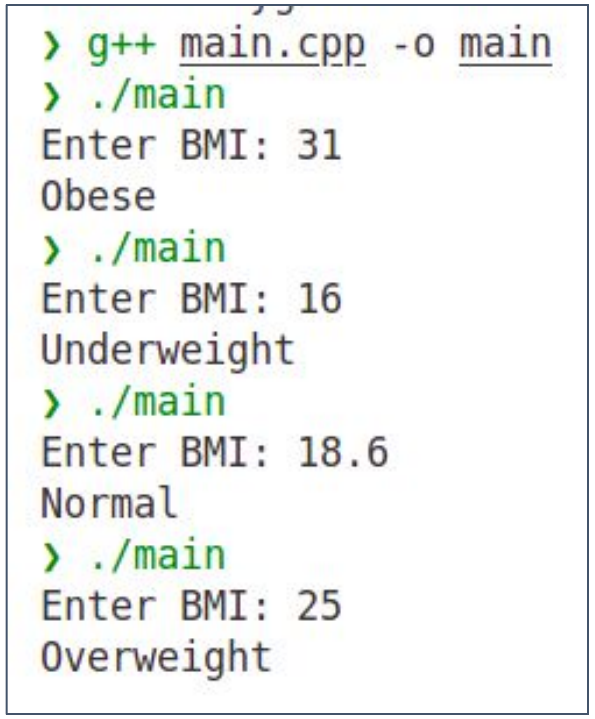
```
❯ ./main
Enter patient's SpO2 (%): 85
Alert: Hypoxemia detected (SpO2 < 90%).
❯ ./main
Enter patient's SpO2 (%): 95
Normal oxygen level.
```

Example of an Linear Nested Selection



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

int main()
{
    float bmi;
    cout << "Enter BMI: ";
    cin >> bmi;

    if (bmi >= 30)
        cout << "Obese\n";
    else if (bmi >= 25)
        cout << "Overweight\n";
    else if (bmi >= 18.5)
        cout << "Normal\n";
    else
        cout << "Underweight\n";
    return 0;
}
```

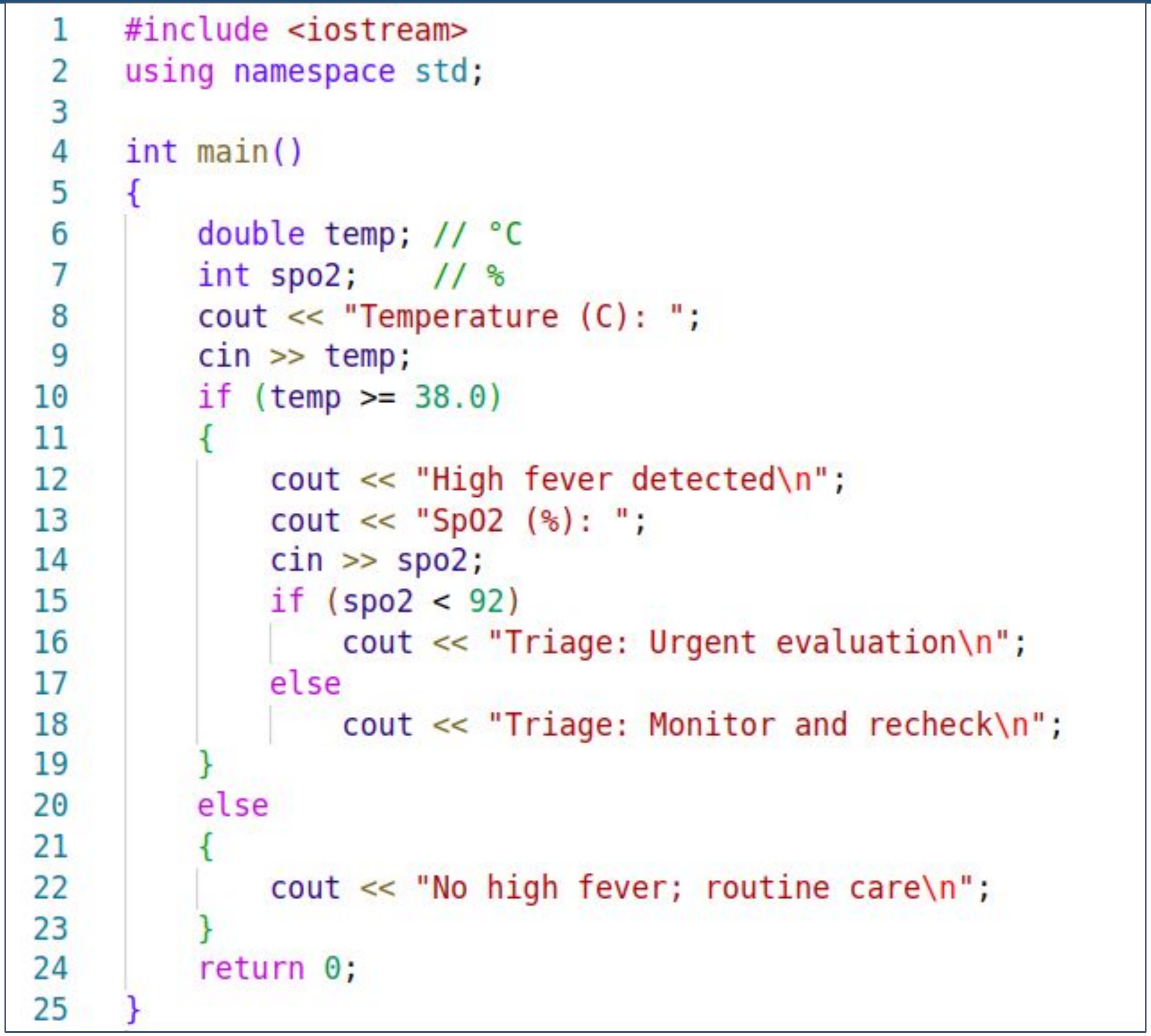
BMI Classification



```
❯ g++ main.cpp -o main
❯ ./main
Enter BMI: 31
Obese
❯ ./main
Enter BMI: 16
Underweight
❯ ./main
Enter BMI: 18.6
Normal
❯ ./main
Enter BMI: 25
Overweight
```

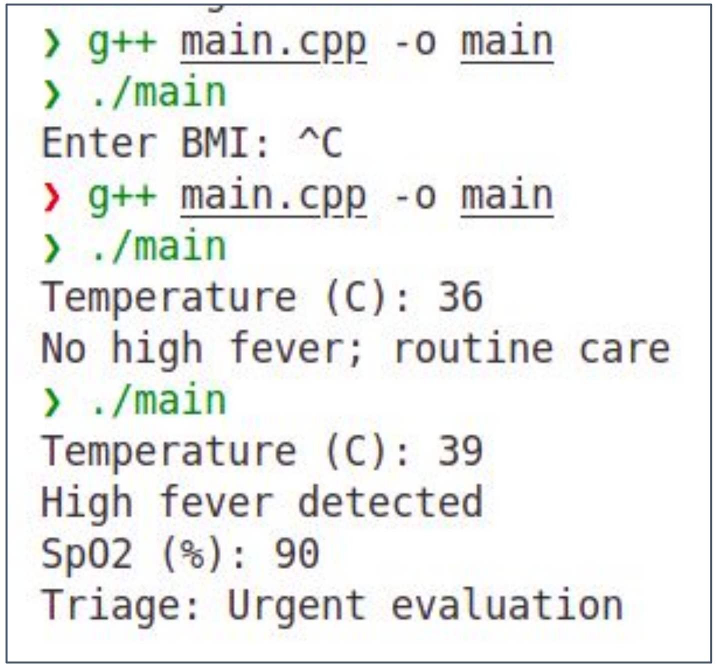
Example of an Non-Linear Nested Selection

UNIVERSITAS GADJAH MADA



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

int main()
{
    double temp; // °C
    int spo2;    // %
    cout << "Temperature (C): ";
    cin >> temp;
    if (temp >= 38.0)
    {
        cout << "High fever detected\n";
        cout << "SpO2 (%): ";
        cin >> spo2;
        if (spo2 < 92)
            cout << "Triage: Urgent evaluation\n";
        else
            cout << "Triage: Monitor and recheck\n";
    }
    else
    {
        cout << "No high fever; routine care\n";
    }
    return 0;
}
```

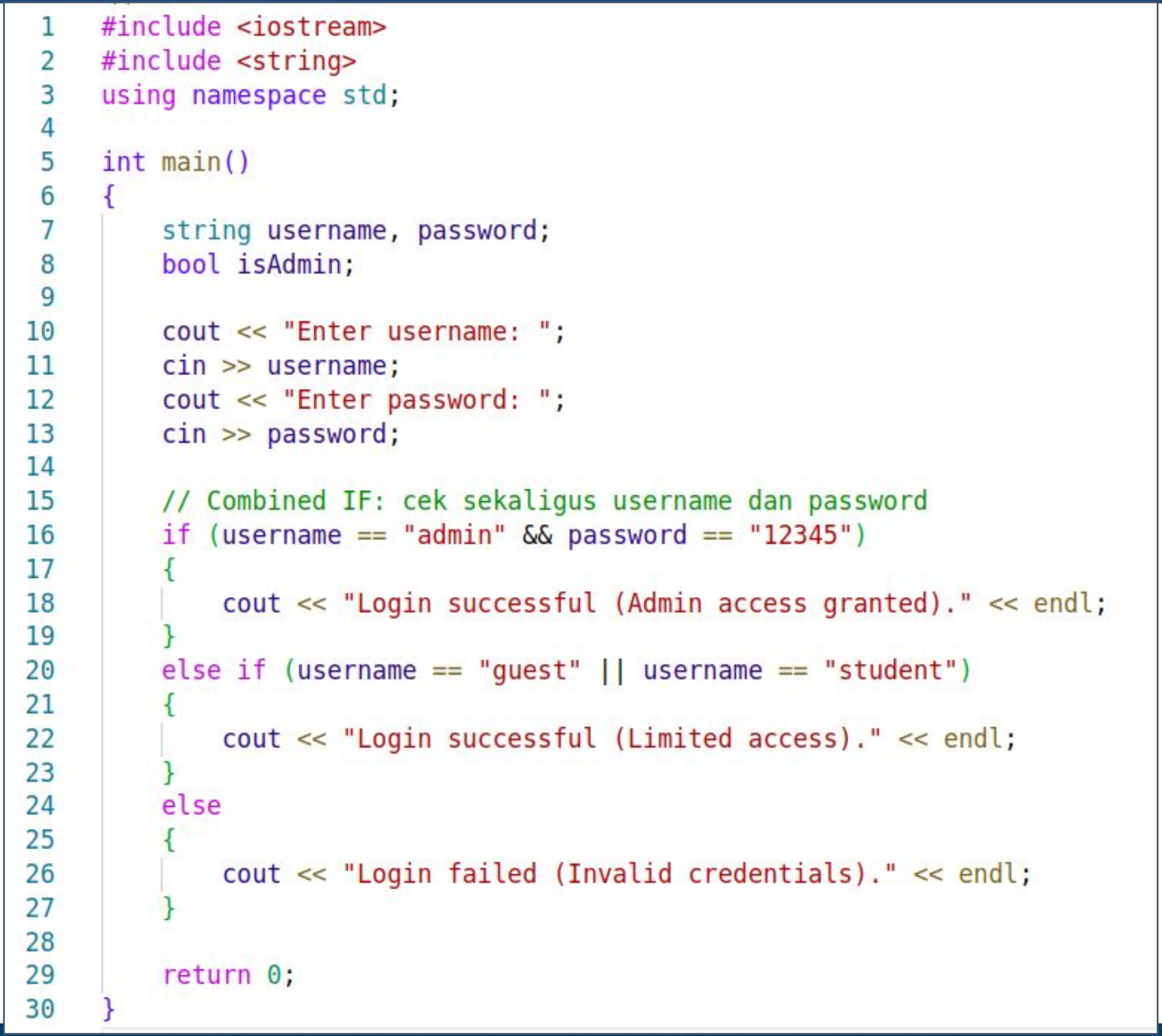
Simple triage of fever & SpO₂



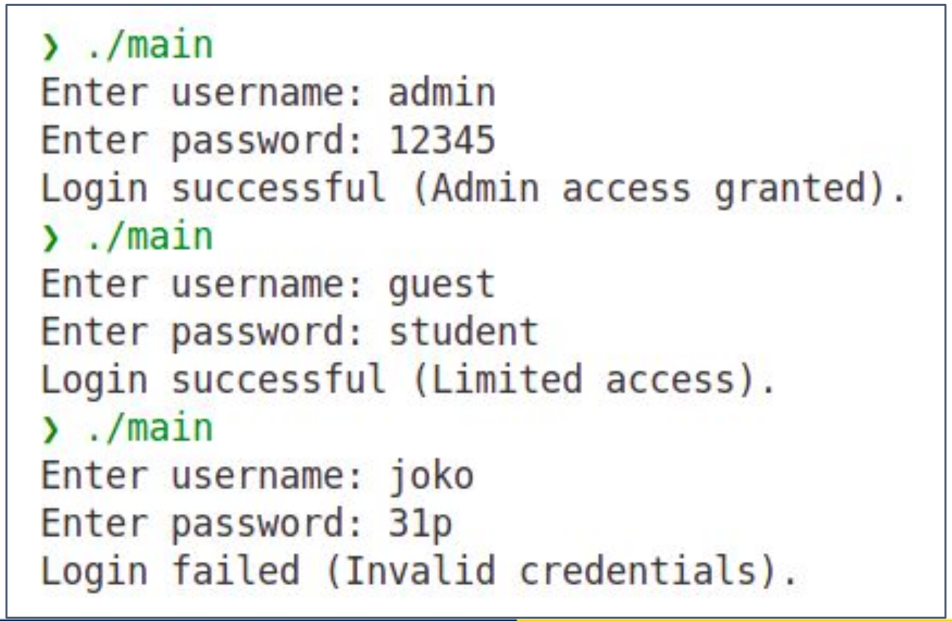
```
❯ g++ main.cpp -o main
❯ ./main
Enter BMI: ^C
❯ g++ main.cpp -o main
❯ ./main
Temperature (C): 36
No high fever; routine care
❯ ./main
Temperature (C): 39
High fever detected
SpO2 (%): 90
Triage: Urgent evaluation
```

Example of an Combined IF: Login App

UNIVERSITAS GADJAH MADA



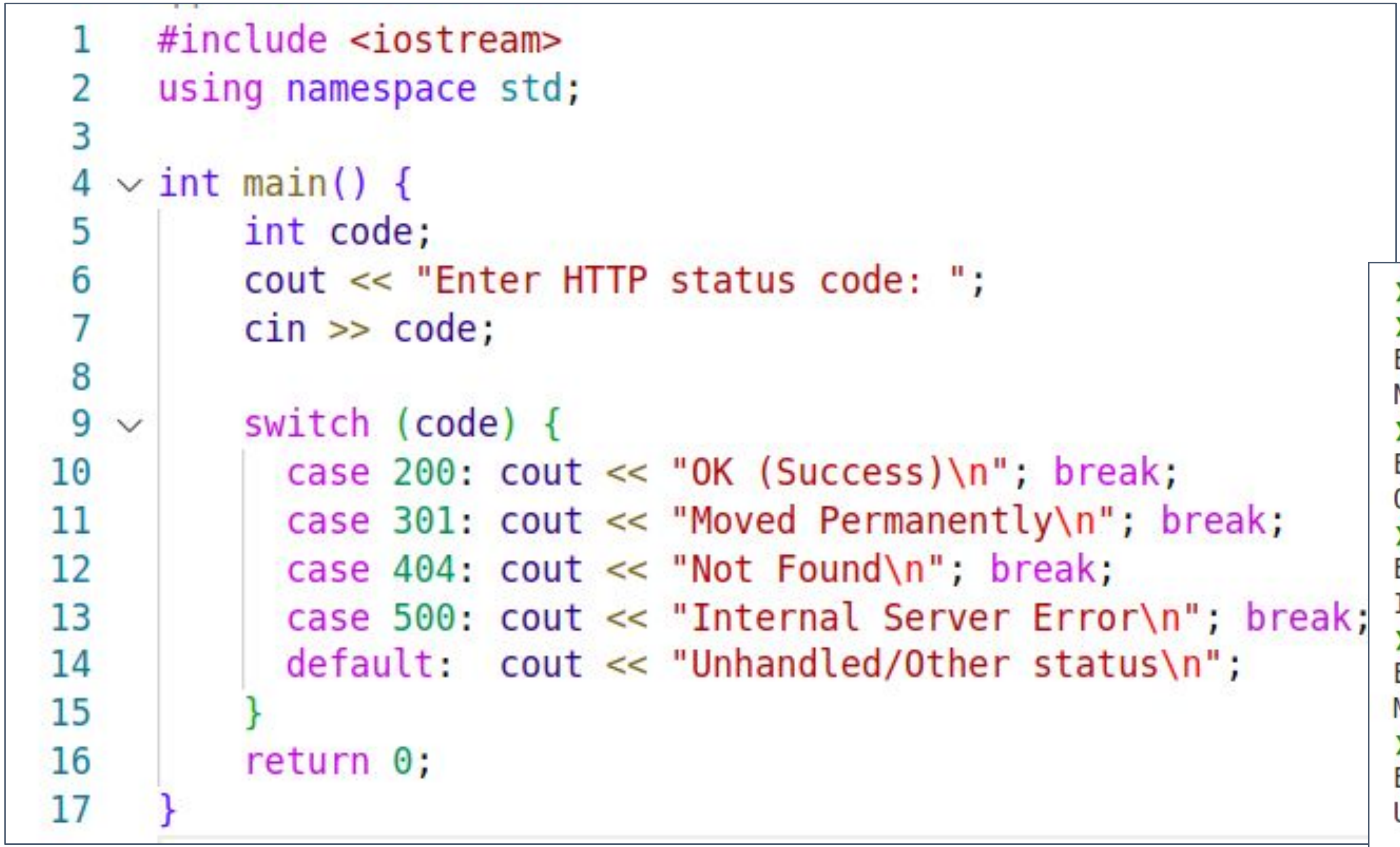
```cpp
#include <iostream>
#include <string>
using namespace std;

int main()
{
    string username, password;
    bool isAdmin;

    cout << "Enter username: ";
    cin >> username;
    cout << "Enter password: ";
    cin >> password;

    // Combined IF: cek sekaligus username dan password
    if (username == "admin" && password == "12345")
    {
        cout << "Login successful (Admin access granted)." << endl;
    }
    else if (username == "guest" || username == "student")
    {
        cout << "Login successful (Limited access)." << endl;
    }
    else
    {
        cout << "Login failed (Invalid credentials)." << endl;
    }

    return 0;
}
```



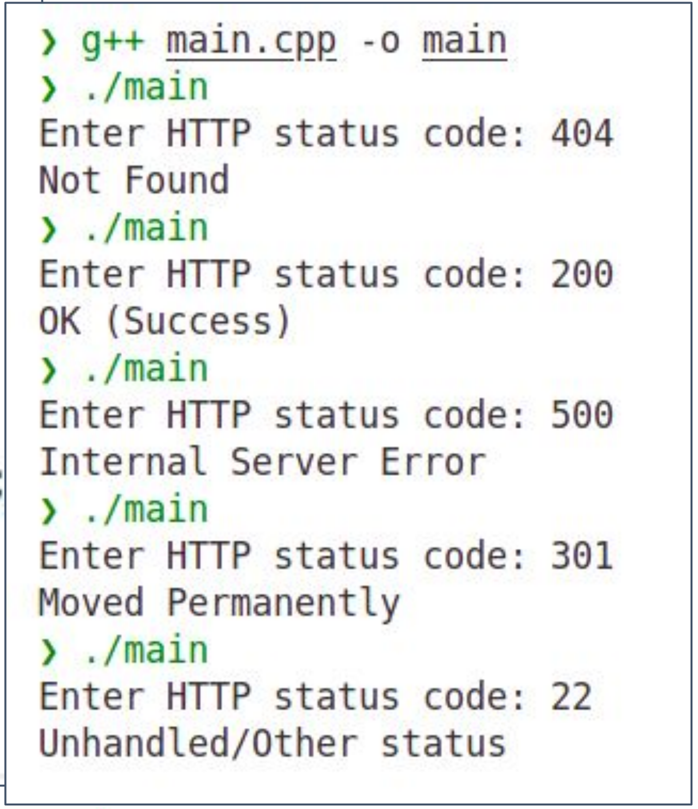
```
❯ ./main
Enter username: admin
Enter password: 12345
Login successful (Admin access granted).
❯ ./main
Enter username: guest
Enter password: student
Login successful (Limited access).
❯ ./main
Enter username: joko
Enter password: 31p
Login failed (Invalid credentials).
```

Example of an Switch Case: HTTP Status Mapper

UNIVERSITAS GADJAH MADA

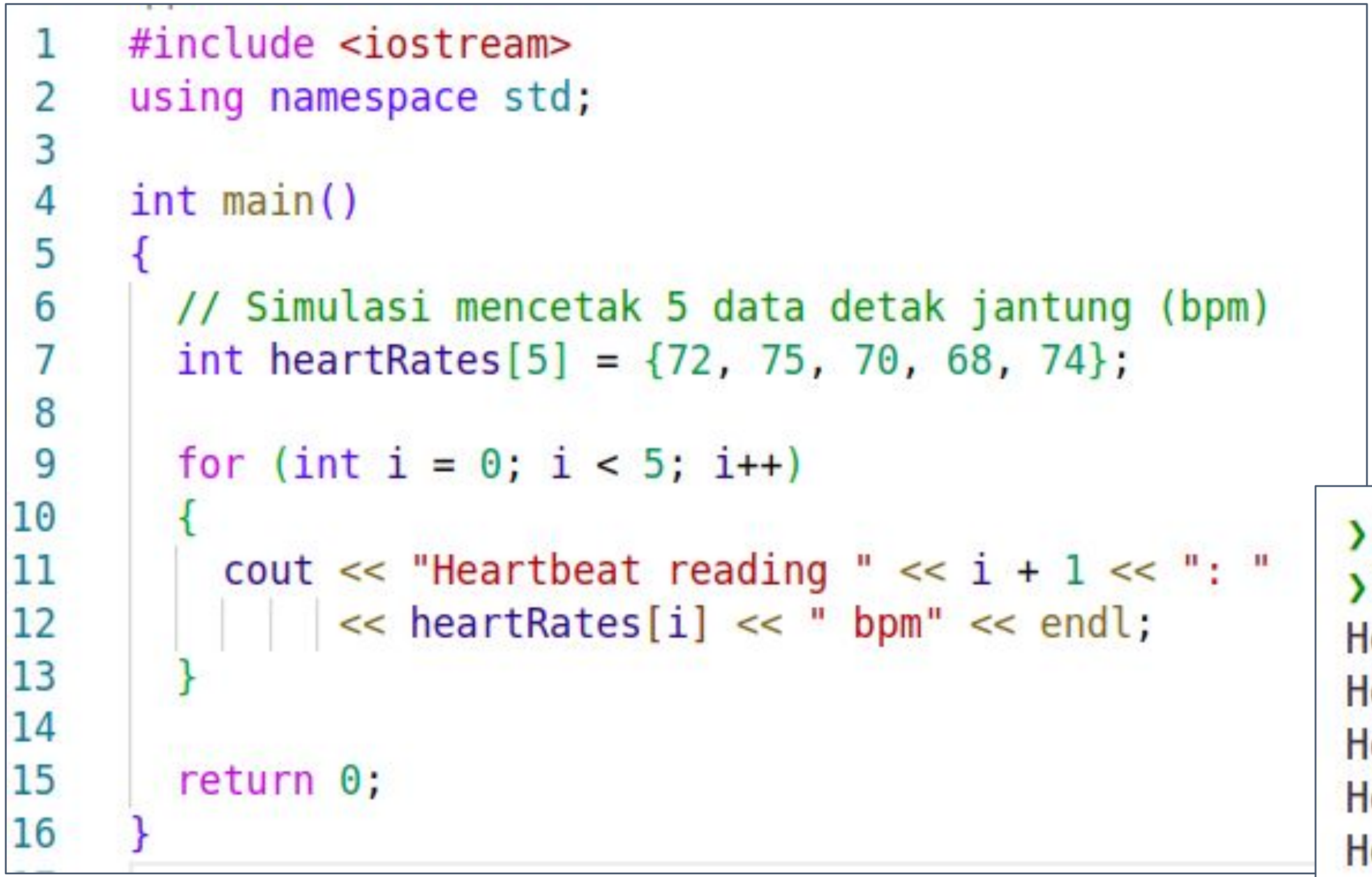


```cpp
#include <iostream>
using namespace std;

int main() {
    int code;
    cout << "Enter HTTP status code: ";
    cin >> code;

    switch (code) {
      case 200: cout << "OK (Success)\n"; break;
      case 301: cout << "Moved Permanently\n"; break;
      case 404: cout << "Not Found\n"; break;
      case 500: cout << "Internal Server Error\n"; break;
      default:  cout << "Unhandled/Other status\n";
    }
    return 0;
}
```

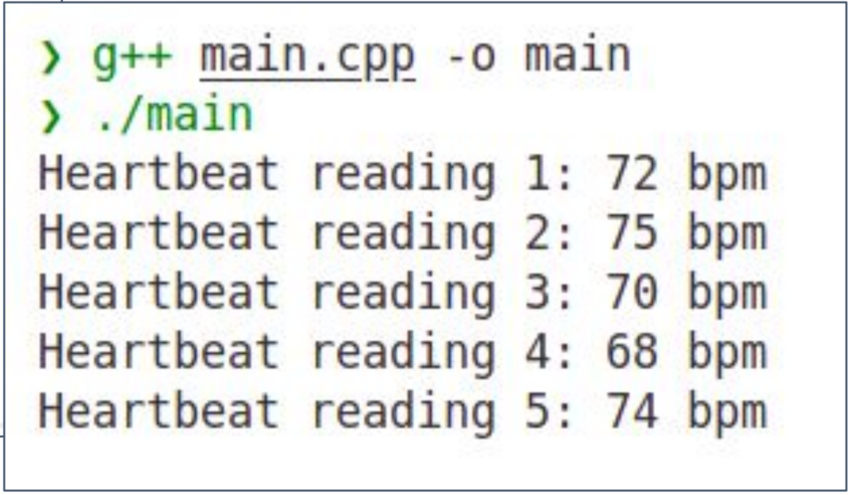


```
❯ g++ main.cpp -o main
❯ ./main
Enter HTTP status code: 404
Not Found
❯ ./main
Enter HTTP status code: 200
OK (Success)
❯ ./main
Enter HTTP status code: 500
Internal Server Error
❯ ./main
Enter HTTP status code: 301
Moved Permanently
❯ ./main
Enter HTTP status code: 22
Unhandled/Other status
```

Example of an For Loop: Heartbeat Data Simulation



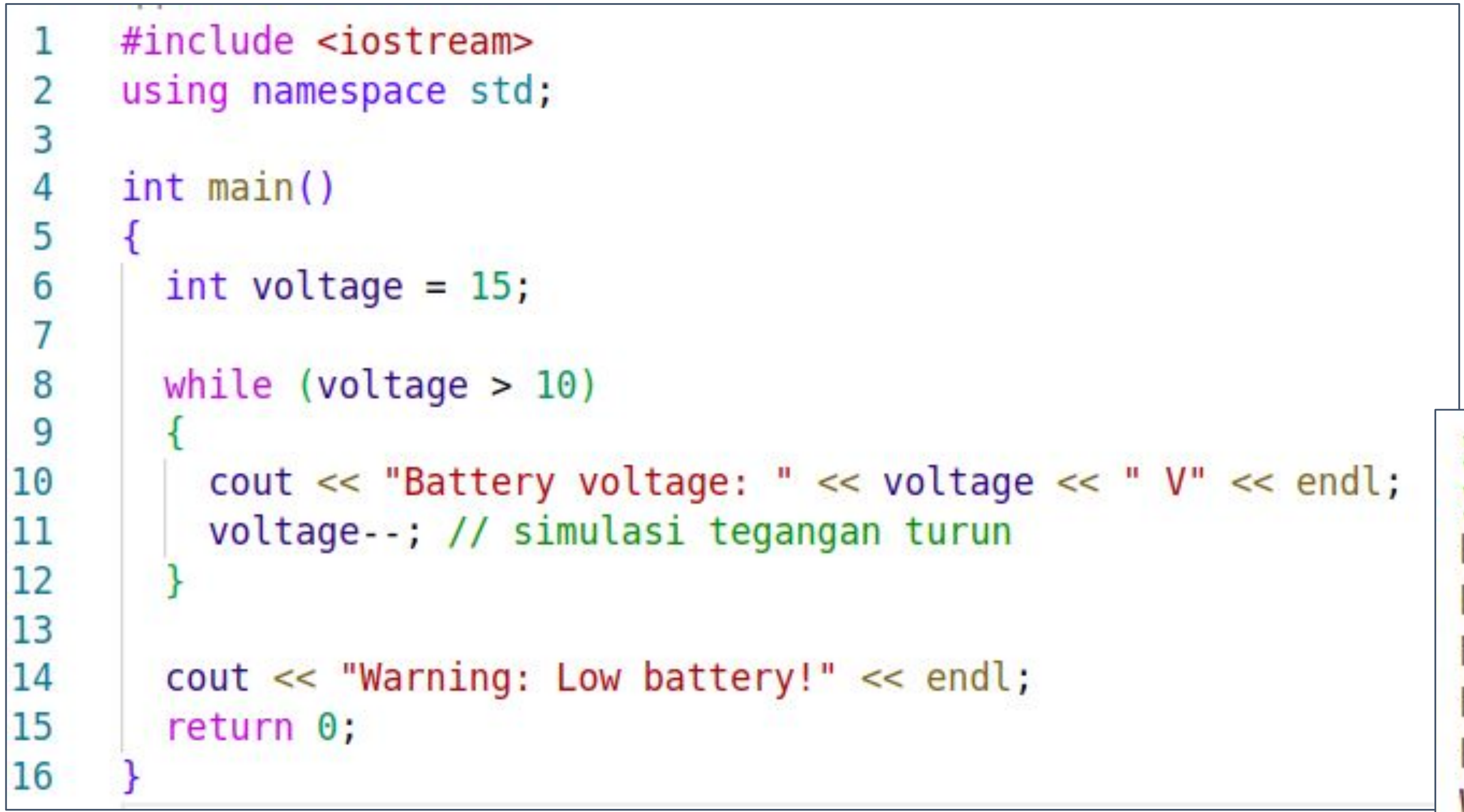
```cpp
#include <iostream>
using namespace std;

int main()
{
    // Simulasi mencetak 5 data detak jantung (bpm)
    int heartRates[5] = {72, 75, 70, 68, 74};

    for (int i = 0; i < 5; i++)
    {
        cout << "Heartbeat reading " << i + 1 << ": "
             << heartRates[i] << " bpm" << endl;
    }

    return 0;
}
```



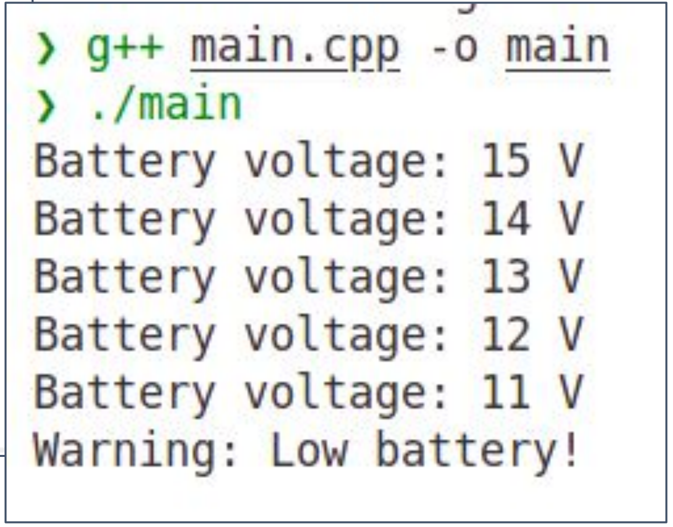
```
❯ g++ main.cpp -o main
❯ ./main
Heartbeat reading 1: 72 bpm
Heartbeat reading 2: 75 bpm
Heartbeat reading 3: 70 bpm
Heartbeat reading 4: 68 bpm
Heartbeat reading 5: 74 bpm
```

Example of an While Loop: Battery Monitoring

UNIVERSITAS GADJAH MADA

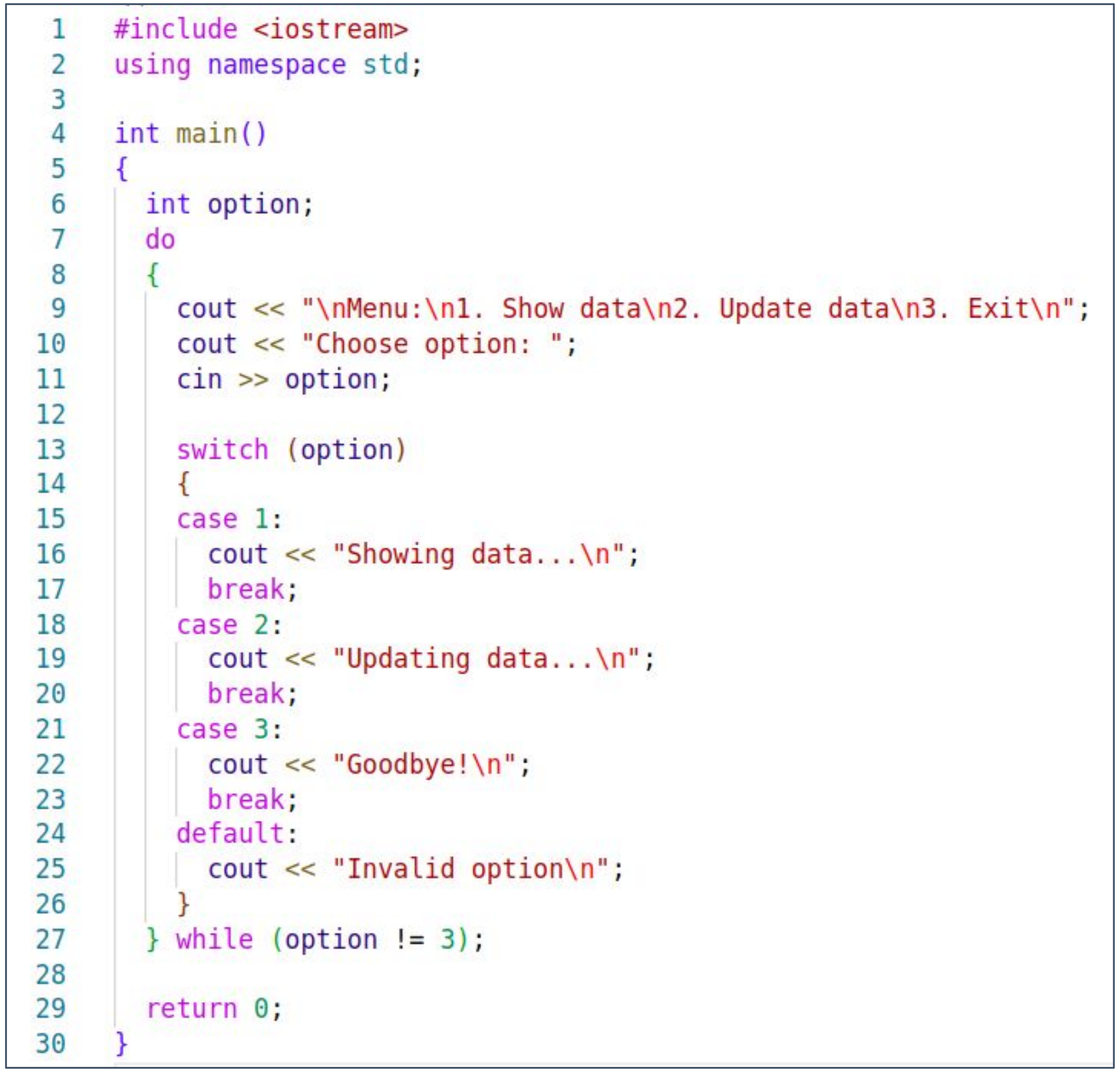


```cpp
#include <iostream>
using namespace std;

int main()
{
    int voltage = 15;

    while (voltage > 10)
    {
        cout << "Battery voltage: " << voltage << " V" << endl;
        voltage--; // simulasi tegangan turun
    }

    cout << "Warning: Low battery!" << endl;
    return 0;
}
```

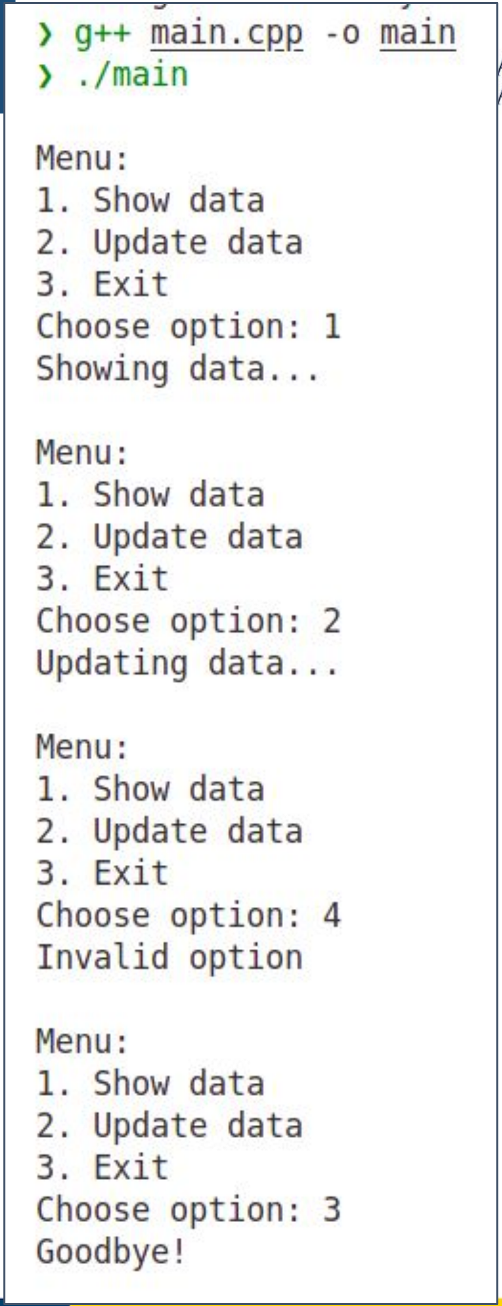


```
❯ g++ main.cpp -o main
❯ ./main
Battery voltage: 15 V
Battery voltage: 14 V
Battery voltage: 13 V
Battery voltage: 12 V
Battery voltage: 11 V
Warning: Low battery!
```

Example of an Do-While Loop: User Menu



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

int main()
{
  int option;
  do
  {
    cout << "\nMenu:\n1. Show data\n2. Update data\n3. Exit\n";
    cout << "Choose option: ";
    cin >> option;

    switch (option)
    {
    case 1:
      cout << "Showing data...\n";
      break;
    case 2:
      cout << "Updating data...\n";
      break;
    case 3:
      cout << "Goodbye!\n";
      break;
    default:
      cout << "Invalid option\n";
    }
  } while (option != 3);

  return 0;
}
```



```
❯ g++ main.cpp -o main
❯ ./main

Menu:
1. Show data
2. Update data
3. Exit
Choose option: 1
Showing data...

Menu:
1. Show data
2. Update data
3. Exit
Choose option: 2
Updating data...

Menu:
1. Show data
2. Update data
3. Exit
Choose option: 4
Invalid option

Menu:
1. Show data
2. Update data
3. Exit
Choose option: 3
Goodbye!
```

UNIVERSITAS GADJAH MADA

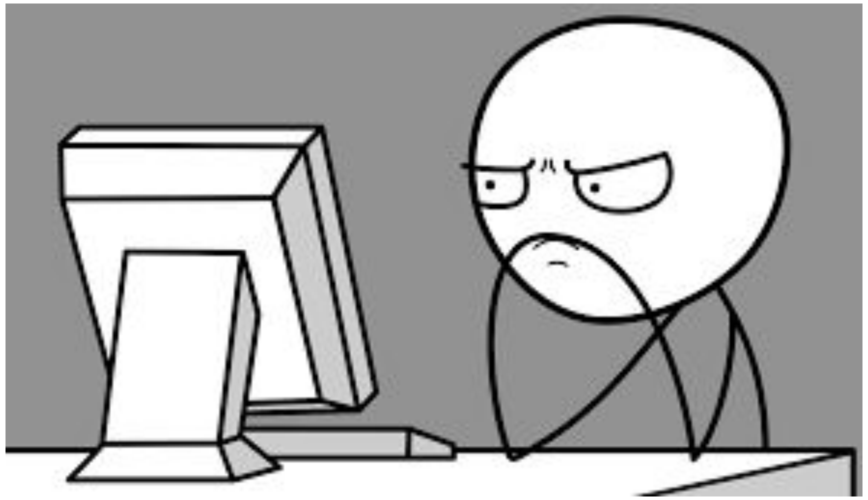
Exercise

Exercise (1)

UNIVERSITAS GADJAH MADA

Create a mathematical expression that can be used to test the condition of the following IF statement (....).

1. age between 12 to 17 years old
   - example: 12 < age < 17
2. water less than 1.5 and more than 0.1
3. year can be divided by 4
4. speed not greater than 55
5. y is greater than x and less than z
6. w is equal to 6 or not more than 3

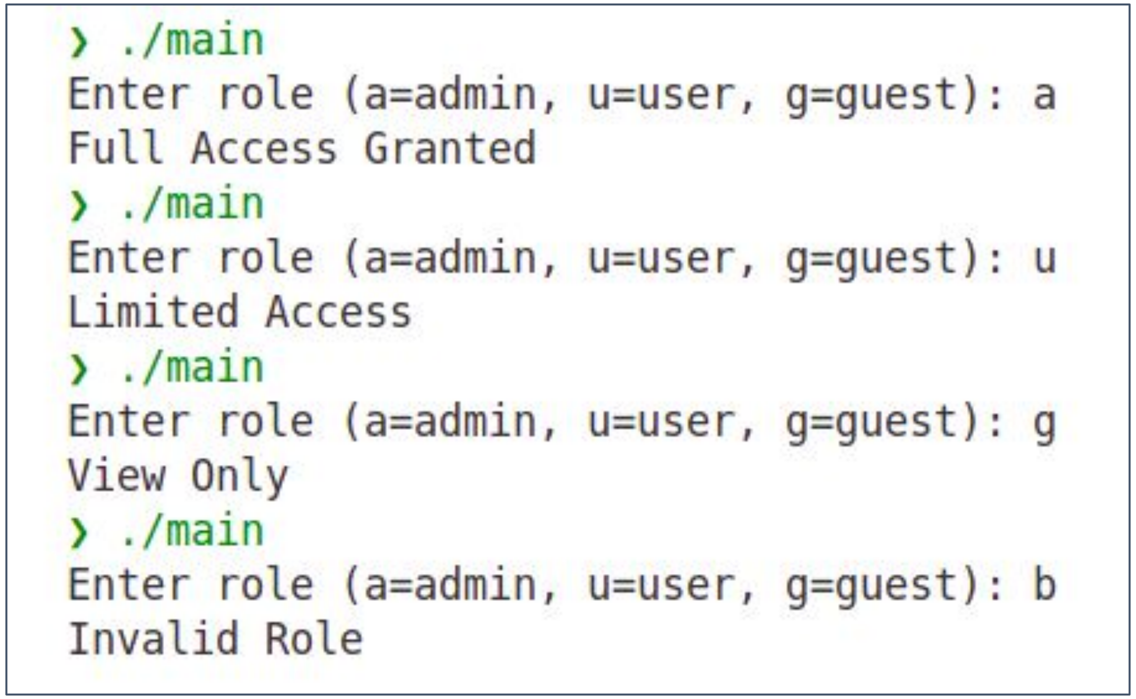


Exercise (2) : Login Role Check

Write a C++ program that asks for a user role (admin, user, guest):

- If role = “admin” → print “Full Access Granted”.
- If role = “user” → print “Limited Access”.
- If role = “guest” → print “View Only”.
- Else → print “Invalid Role”.

sample output



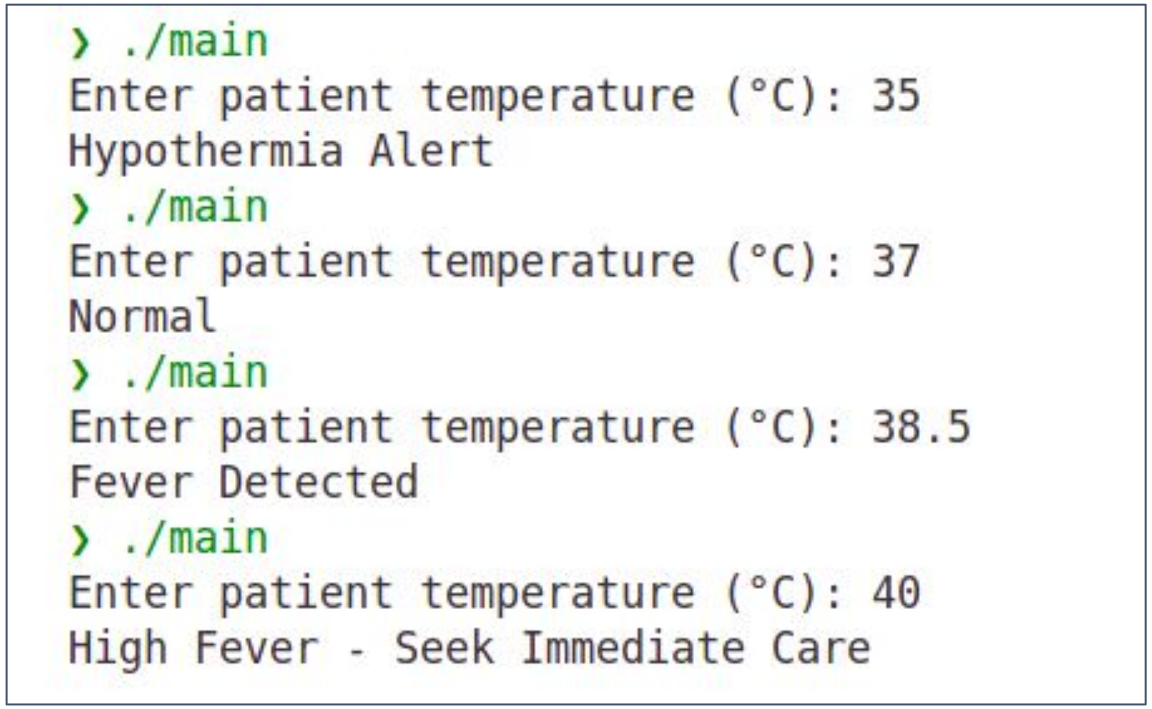
```
❯ ./main
Enter role (a=admin, u=user, g=guest): a
Full Access Granted
❯ ./main
Enter role (a=admin, u=user, g=guest): u
Limited Access
❯ ./main
Enter role (a=admin, u=user, g=guest): g
View Only
❯ ./main
Enter role (a=admin, u=user, g=guest): b
Invalid Role
```

Exercise (3) : Patient Vital Check (Temperature)

Write a C++ program that checks patient's body temperature:

- Temp < 36°C → print "Hypothermia Alert".
- 36–37.5°C → print "Normal".
- 37.5°C and < 39°C → print "Fever Detected".
- ≥ 39°C → print "High Fever – Seek Immediate Care".

sample output



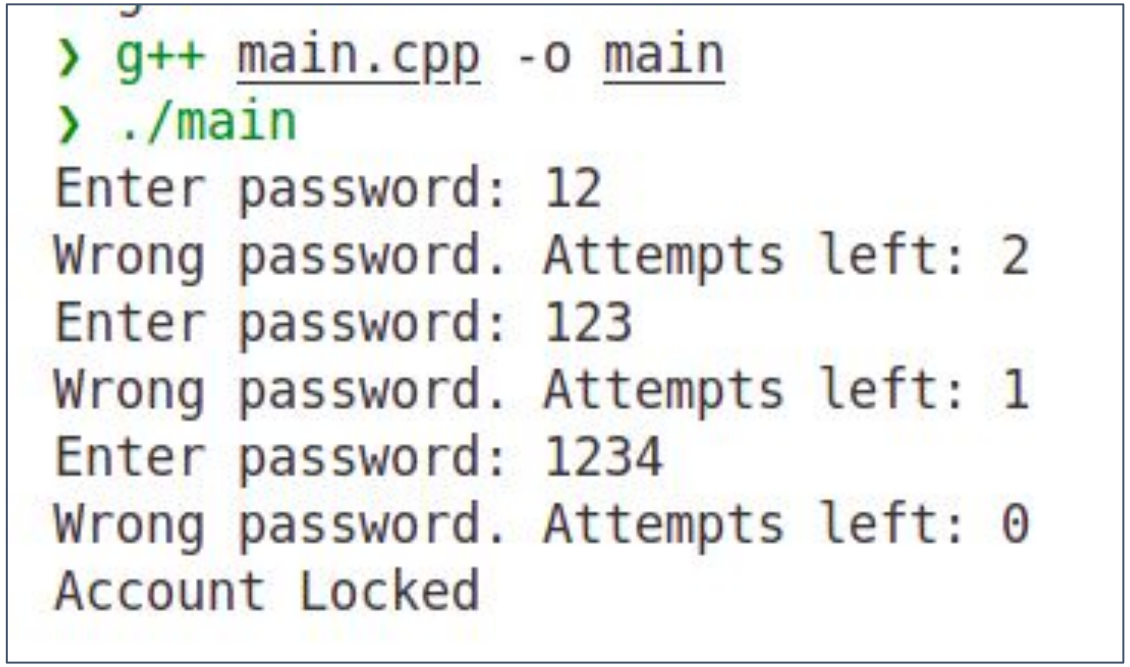
```
❯ ./main
Enter patient temperature (°C): 35
Hypothermia Alert
❯ ./main
Enter patient temperature (°C): 37
Normal
❯ ./main
Enter patient temperature (°C): 38.5
Fever Detected
❯ ./main
Enter patient temperature (°C): 40
High Fever - Seek Immediate Care
```

Exercise (4) : Login Attempts

Write a C++ program using a **for/while loop**:

- User has **3 attempts** to enter password.
- If correct → print "Access Granted" and exit.
- If wrong after 3 attempts → print "Account Locked".

sample output



```
❯ g++ main.cpp -o main
❯ ./main
Enter password: 12
Wrong password. Attempts left: 2
Enter password: 123
Wrong password. Attempts left: 1
Enter password: 1234
Wrong password. Attempts left: 0
Account Locked
```

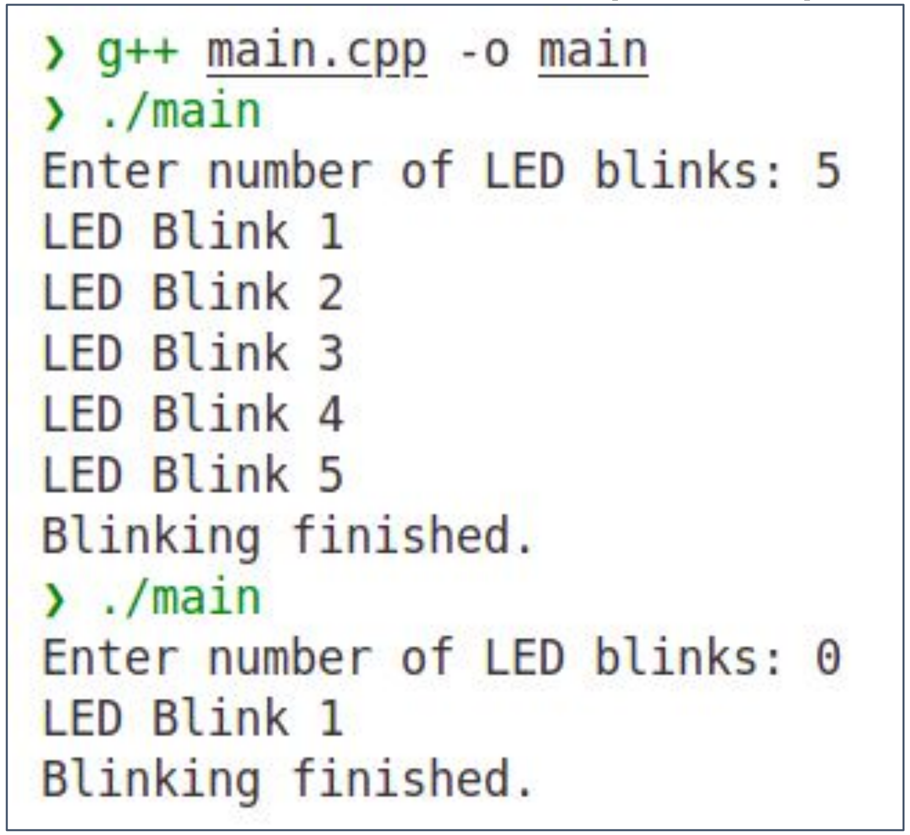
Exercise (5) : LED Blinking Simulation

UNIVERSITAS GADJAH MADA

Write a C++ program using a **do–while loop**:

- Simulate LED blinking.
- User inputs number of blinks.
- Program must blink at least once, even if user enters 0.

sample output



```
❯ g++ main.cpp -o main
❯ ./main
Enter number of LED blinks: 5
LED Blink 1
LED Blink 2
LED Blink 3
LED Blink 4
LED Blink 5
Blinking finished.
❯ ./main
Enter number of LED blinks: 0
LED Blink 1
Blinking finished.
```

UNIVERSITAS
GADJAH MADA

Thank You!
See you, next week, stay safe!