Fundamentals of C++ Programming

Control Structures (part 1)
Lecturer: Duc Dung Nguyen
Credits: 4
Outcomes

- Understand basic control structures in C/C++
  - if-else statement
  - switch statement
- Solve the problem using conditional executions
- Implement if-else, switch-case statements
Today’s outline

- Conditional execution
- if-else statement
  - Nested conditionals
- switch statement
  - Enum type
Conditional execution
Conditional execution

- Boolean expression: evaluate to true / false
- What is true? What is false?
- bool type
- Type conversion
- Assignment
- Common expressions
Conditional execution

- **Type** `bool`: true / false
  - Size: 1 byte (basic unit of storage)
  - Be represented as integer: true = 1, false = 0
- What happens when you assign a value to boolean type:
  - **False**: 0 value (for integer, floating point number, character ‘\0’)
  - **True**: anything else (except structures, unless a casting operator is defined)
## Conditional execution

- **Relational operators**

<table>
<thead>
<tr>
<th>Operator</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;==&quot;</td>
<td>Equal to</td>
</tr>
<tr>
<td>&quot;&lt;&quot;</td>
<td>Less than</td>
</tr>
<tr>
<td>&quot;&gt;&quot;</td>
<td>Greater than</td>
</tr>
<tr>
<td>&quot;&lt;=&quot;</td>
<td>Less than or equal to</td>
</tr>
<tr>
<td>&quot;&gt;=&quot;</td>
<td>Greater than or equal to</td>
</tr>
<tr>
<td>&quot;!=&quot;</td>
<td>Not equal to</td>
</tr>
</tbody>
</table>
Conditional execution

❖ Logic operators

<table>
<thead>
<tr>
<th>Operator</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>!</td>
<td>not</td>
</tr>
<tr>
<td>&amp;&amp;</td>
<td>and</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

CuuDuongThanCong.com

https://fb.com/tailieudientucntt
Conditional execution

Examples:

```c
bool b = true, b1 = false;
int a = -1, c = 0;
float x = 0.5f, y = 1.2f;
b = a > c;
b1 = a;
b = c;
b1 = x < y && a > c;
b = x;
c = y + b1;
b1 = 50 != 'a';
b = x + 4.9 < y / 0.5f;
```
If-else statement
If-else statement

❖ Simple if statement:

❖ Execute a statement or a list of statements if the given condition is satisfied

❖ `if (<conditional expression>) <statement>;

❖ `if (<conditional expression>) {
    <statements>
}

❖ E.g.:

❖ `if (a > b)
    cout << a << " is greater than " << b << endl;`
**If-else statement**

- **Flowchart**
If-else statement

● Example:

```cpp
#include <iostream>

int main() {
    float a, b, c, delta;
    cout << "Please input three real values a, b, c: ";
    cin >> a >> b >> c;
    delta = b * b - 4 * a * c;
    if (delta < 0)
        cout << "Have no real root" << endl;
    return 0;
}
```

```
#include <iostream>

int main() {
    float a, b, c, delta;
    cout << "Please input three real values a, b, c: ";
    cin >> a >> b >> c;
    delta = b * b - 4 * a * c;
    if (delta < 0)
        cout << "Have no real root" << endl;
    return 0;
}
```

CuuDuongThanCong.com
https://fb.com/tailieudientucntt
If-else statement

❖ Examples

```cpp
#include <iostream>

int main() {
    float a, b, c, delta;
    cout << "Input three real values a, b, c: ";
    cin >> a >> b >> c;
    delta = b * b - 4 * a * c;
    if (delta < 0) {
        cout << "Delta value is negative" << endl;
        cout << "Have no real root" << endl;
    }
    return 0;
}
```
If-else statement

❖ Full if-else statement:

❖ if (<conditional expression>) <if-true statement>; else <if-false statement>;
❖ if (<conditional expression>) {
      <if-true statements>
    } else {
      <if-false statements>
    }
If-else statement

❖ Flowchart
If-else statement

Examples

```cpp
#include <iostream>

int main() {
    float a, b, c, delta;
    cout << "Input three real values a, b, c: ";
    cin >> a >> b >> c;
    delta = b * b - 4 * a * c;
    if (delta < 0) {
        cout << "Delta value is negative" << endl;
        cout << "Have no real root" << endl;
    } else cout << "The quadratic equation has at least one root" << endl;
    return 0;
}
```
Nested conditionals

❖ Nested if-else statements

❖ if (<exp>) // first check
  if (<exp>) // second check
    if (<exp>) // third check
     <statement>
  else <statement>
else <statement>
else if (<exp>) <statement>
else if (<exp>) <statement>
else <statement>
Nested conditionals

❖ Nested if-else statements: multi-way

❖ if (<exp 1>) <statement 1>
else if (<exp 2>) <statement 2>
else if (<exp 3>) <statement 3>
else <statement 4>

❖ if (<exp 1>) <statement 1>
else if (<exp 2>) <statement 2>
    else if (<exp 3>) <statement 3>
    else <statement 4>
Switch statement
Switch statement

❖ A convenient way to write multi-way statement

❖ `switch(<exp>) {`
    ```
    case <value 1>: <statements>;
    case <value 2>: <statements>;
    ...
    case <value N>: <statements>;
    default: <statements>;
    }
    ```
Switch statement

❖ Flowchart

Switch statement

- <case1>
  - Y: case 1 statements
  - N: <case2>
    - Y: case 2 statements
    - N: <case3>
      - Y: case 3 statements
      - N: defaults

- statement

[Source: cuu.duong.than.cong.com]
Switch statement

❖ A convenient way to write multi-way statement

❖ `switch(<exp>) {
  case <value 1>: <statements>; break;
  case <value 2>: <statements>; in break;
  ...
  case <value N>: <statements>; break;
  default: <statements>;
}

CuuDuongThanCong.com
https://fb.com/tailieudientucntt
Switch statement

Example:

```cpp
#include <iostream>
int main() {
    // Do something
    int usrChoice;
    cout << "Please make your choice (1~4): ";
    cin >> usrChoice;
    if (usrChoice < 1 || usrChoice > 4)
        cout << "Cannot recognise your choice" << endl;
    else switch (usrChoice) {
        case 1:
            cout << "pick 1" << endl;
            cout << "action 1" << endl;
        case 2:
            cout << "pick 2" << endl;
            cout << "action 2" << endl;
        case 3:
            cout << "pick 3" << endl;
            cout << "action 3" << endl;
        default:
            cout << "pick 4" << endl;
            cout << "action 4" << endl;
    }
    return 0;
}
```

```cpp
#include <iostream>
int main() {
    // Do something
    int usrChoice;
    cout << "Please make your choice (1~4): ";
    cin >> usrChoice;
    switch (usrChoice) {
        case 1:
            cout << "pick 1" << endl;
            cout << "action 1" << endl; break;
        case 2:
            cout << "pick 2" << endl;
            cout << "action 2" << endl; break;
        case 3:
            cout << "pick 3" << endl;
            cout << "action 3" << endl; break;
        case 4:
            cout << "pick 4" << endl;
            cout << "action 4" << endl; break;
        default:
            cout << "Cannot recognise your choice" << endl;
    }
    return 0;
}
```
Enumerated type

- Define a list of possible values of a type
  - `enum <type name> {<name of possible values>};`
  - `enum [<type name>] {<name of possible values>} <variables>;

- Example:
  - `enum Color {Red, Orange, Yellow, Green, Blue, Violet};
    Color c = Yellow;
    out << "Yellow color has value: " << c << endl;`
Enumerated type

- Define a list of possible values of a type
  - `enum <type name> {<name0 = value0>, <name1 = value1>, ...};`
  - `enum [<type name>] {<name0>};`

- Example:
  - `enum Color {Red = -1, Orange = 2, Yellow = 8, Green = 3, Blue, Violet};`
  - `Color c = Blue;`
  - `out << "Blue color has value: " << c << endl;`
Enumerated type and switch statement

```cpp
#include <iostream>
enum usrChoice {Accept = 1, Denied = 0, Undecided = -1}
int main() {
    usrChoice choice;
    cout << " 1: Accept" << endl
    << " 0: Denied" << endl
    << "-1: have not decided yet" << endl
    << "Please make your choice: ";
cin >> choice;
switch (choice) {
    case Accept:
        cout << "Thank you for accepting my document." << endl; break;
    case Denied:
        cout << "I’m sorry for hearing that." << endl; break;
    case Undecided:
        cout << "Please call me again when you make decision" << endl; break;
    default:
        cout << "You may have another choice but I cannot understand it." << endl;
}
return 0;
}
```
If-else vs. switch

❖ switch statement can be represented by a sequence of if-else statements
❖ if-else statement is more general
❖ All control structures can be represented by if-else statements and goto statements (with the support from labels).
Conditional operator

- Syntax:
  - `<expression> ? <if-true expression> : <if-false expression>`
  - Equivalent to if-else statement but apply for expressions

- Example:
  - `char outChar;
    outChar = a == 'c' ? 'C' : 'c';`
  - `float diff;
    diff = x > y ? x - y : y - x;`
Conditional operator

- **Safety**: unless you understand how your code works (including compiler mechanism), write safe code.

- Example:
  - `char outChar = (a == 'c')? 'C': 'c';`
  - `float diff = (x > y)? (x - y): (y - x);`
  - `float r = ((a ^ b) & 0x0010)? (x - (a == b? 1.5: -1.5))): (y - x);`
Preprocessor directives
Preprocessor directives

❖ Conditional Pre-processor directives:

❖ `#define`, `#undef`, `#ifdef`, `#ifndef`, `#else`, `#elif`, `#endif`

❖ E.g.:

❖ `int foo(float a, double b) {
   #ifdef ___MSC_VER
      return a * 3.14159 - sqrt(b * a);
   #else
      return a * 3.14159 + b * b;
   #endif
}
`
Preprocessor directives

♦ Conditional Pre-processor directives

♦ Library headers (*.h, *.hpp):

♦ #pragma once
   // library definition

♦ #ifndef __MY_LIBRARY_H__
#define __MY_LIBRARY_H__

♦ #ifdef __MY_LIBRARY_H__
   // library definition
#endif
Preprocessor directives

- Power of macros and preprocessor directive
  - One definition fit all
  - Flexible, portable
  - Open source community
Summarise

- Understand basic elements of C/C++
- Principle of conditional execution
- if-else statement, nested conditionals
- switch statement
- Conditional operator
Quiz & homework

- Use if-else statement to write the program that asks user to input three real values. Compare those values and tell user, which number is the maximum, minimum, and median.

- Write a program to solve the quadratic equation. Tell user all possible roots, including complex roots.