

VÕ TIẾN

Thảo luận kiến thức CNTT trường BK về KHMT(CScience), KTMT(CEngineering)
<https://www.facebook.com/groups/khmt.ktmt.cse.bku>



Kỹ Thuật Lập Trình (Cơ bản và nâng cao C++)

KTILT1 - HK242

TASK 2 NMLT và cách debug

Thảo luận kiến thức CNTT trường BK
về KHMT(CScience), KTMT(CEngineering)
<https://www.facebook.com/groups/khmt.ktmt.cse.bku>

Mục lục

1	Phong cách code và debug	2
1.1	Cách Format Code	2
1.2	Debug	3
1.3	Bài Tập	6



1 Phong cách code và debug

1.1 Cách Format Code

Dấu ngoặc nhọn {}

- Mở dấu { cùng dòng với câu lệnh, không xuống dòng.
- Đóng dấu } luôn thẳng hàng với khối mở.

Đúng

```
1 if (condition) {
2     doSomething();
3 } else {
4     doSomethingElse();
5 }
```

Sai

```
1 if (condition)
2 {
3     doSomething();
4 }
```

Indentation (Thụt lề)

- Dùng 4 khoảng trắng (spaces), không dùng tab.
- Mỗi cấp độ lồng nhau phải thụt vào 4 spaces.

Đúng

```
1 for (int i = 0; i < 10; i++) {
2     if (arr[i] > 0) {
3         process(arr[i]);
4     }
5 }
```

Sai

```
1 for (int i = 0; i < 10; i++)
2 {
3     if (arr[i] > 0)
4     {
5         process(arr[i]);
6     }
7 }
```

Khai báo #include

- Sắp xếp theo thứ tự:
 1. Thư viện chuẩn (<iostream>, <vector>, <cmath>, ...).
 2. Thư viện bên thứ ba (nếu có).
 3. File header tự định nghĩa ("MyHeader.h").
- Không include thừa.

Tên biến

- Sử dụng camelCase cho biến và phương thức.
- Biến hằng số viết UPPER_CASE.

Đúng

```
1 int maxValue;
2 double learningRate;
3 const int MAX_SIZE = 100;
```

Sai

```
1 int max_value; // Dùng snake_case
   → không thống nhất
2 double LearningRate; // Dùng
   → PascalCase cho biến
```

Hàm



- Tên hàm dùng camelCase, không có khoảng trắng trước .
- Khai báo const nếu phương thức không thay đổi dữ liệu.

Đúng

```
1 int computeGradient() const;  
2 void processData(const  
   ↳ std::vector<int>& data);
```

Sai

```
1 int ComputeGradient (); // Dư dấu  
   ↳ cách trước ()  
2 int compute_gradient(); // Dùng  
   ↳ snake_case không thống nhất
```

Định dạng vòng lặp và điều kiện

- Không để dấu ; sau if, for, while.
- Luôn có { } cả khi chỉ có một dòng.
- Có khoảng trắng trước và sau toán tử so sánh (==, !=, <, >, &&, ||).

Đúng

```
1 if (x == 10) {  
2     std::cout << "x is 10\n";  
3 }  
4  
5 for (int i = 0; i < 10; i++) {  
6     process(i);  
7 }
```

Sai

```
1 if (x==10) std::cout << "x is 10\n";  
   ↳ // Không có khoảng trắng và  
   ↳ thiếu {}
```

Định dạng toán tử

- Dùng dấu cách quanh toán tử =, +, -, *, /, &&, ||, ==, !=, <=, >=.
- Không có khoảng trắng sau (hoặc trước).

Đúng

```
1 int sum = a + b;  
2 if (x > 10 && y < 20) {  
3     result = x * y;  
4 }
```

Sai

```
1 int sum=a+b; // Không có khoảng  
   ↳ trắng  
2 if(x>10&&y<20){ // Thiếu dấu cách  
   ↳ và xuống dòng
```

1.2 Debug

Debug là quá trình tìm kiếm, phân tích và sửa lỗi trong chương trình. Đây là một phần quan trọng trong lập trình, giúp đảm bảo chương trình chạy đúng và hiệu quả.

Các Kỹ Thuật

1. Dùng Logging (In giá trị)

- In giá trị ra màn hình để kiểm tra

```
1 std::cout << "x = " << x << std::endl;
```

- Ưu điểm: Đơn giản, dễ làm.



- Nhược điểm: Khó debug khi code lớn, cần sửa code để in log.
2. Dừng Debugger (GDB, LLDB, VS Code) Debug từng dòng, xem giá trị biến, stack trace, v.v.

Dùng Debugger (GDB, LLDB, VS Code)

Debugger là công cụ giúp theo dõi, kiểm tra và sửa lỗi chương trình mà không cần in `std::cout`.

Có nhiều debugger phổ biến:

- **GDB**: Debugger phổ biến nhất trên Linux.
- **LLDB**: Debugger mặc định của macOS, thay thế GDB.
- **VS Code Debugger**: Giao diện đồ họa hỗ trợ cả GDB & LLDB.

Debugger cho phép:

- Đặt **breakpoint** (dừng chương trình tại một dòng).
- Theo dõi giá trị biến khi chương trình chạy.
- Bước qua từng dòng code (Step Over, Step Into, Step Out).
- Kiểm tra **call stack**, xem hàm nào gọi hàm nào.
- Phát hiện lỗi như **Segmentation Fault**, **Memory Leak**.

Hướng dẫn dùng

1. Cài Extensions C/C++ (Microsoft)
2. Kiểm tra `gdb --version`

```
PS C:\Code\Courser\KTLT\debug> gdb --version
GNU gdb (GDB) 15.2
Copyright (C) 2024 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
```

3. Ví dụ đoạn code sau

```
1  #include <iostream>
2
3  int main() {
4      int n;
5      std::cout << "Enter a number: ";
6      std::cin >> n;
7
8      int factorial = 1;
9      for (int i = 1; i <= n; i++) {
10         factorial *= i;
11     }
12
13     std::cout << "Factorial of " << n << " is: " << factorial << std::endl;
14     return 0;
15 }
```

4. Cấu hình trình biên dịch



```
.vscode
- tasks.json
- launch.json

// tasks.json
{
  "version": "2.0.0",
  "tasks": [
    {
      "label": "Build C++",
      "type": "shell",
      "command": "g++",
      "args": ["-g", "main.cpp", "-o", "main.exe"],
      "group": {
        "kind": "build",
        "isDefault": true
      }
    }
  ]
}

// launch.json
{
  "version": "0.2.0",
  "configurations": [
    {
      "name": "Debug C++",
      "type": "cppdbg",
      "request": "launch",
      "program": "${workspaceFolder}/main.exe",
      "args": [],
      "stopAtEntry": false,
      "cwd": "${workspaceFolder}",
      "environment": [],
      "console": "externalTerminal",
      "MIMode": "gdb",
      "setupCommands": [
        {
          "description": "Enable pretty-printing",
          "text": "-enable-pretty-printing",
          "ignoreFailures": true
        }
      ],
      "preLaunchTask": "Build C++"
    }
  ]
}
```

- **"args": ["-g", "main.cpp", "-o", "main.exe"]** biên dịch chương trình tùy vào file bạn muốn biên dịch và tên file được biên dịch ra
- **"program": "\${workspaceFolder}/main.exe"** đường dẫn đến file thực thi

5. Debug code trong VSCode

- Mở tab **"Run and Debug"** (Ctrl+Shift+D)
- Chọn **"Debug C++"**
- Nhấn **"Start Debugging"** (F5)



- (d) **Đặt breakpoint** (Click vào dòng số bên trái)
- (e) Khi chạy, gõ số vào terminal, debug từng bước với:
- F10 (Next)
 - F11 (Step Into)
 - Shift+F5 (Stop)

Mô tả kĩ hơn <https://discord.com/channels/1334472759945990184/1338426563976495115/1338427924772032543>

1.3 Bài Tập

<https://discord.com/channels/1334472759945990184/1338426563976495115>