W05-2021-11-03-Notes

Lỗi uninitialized before used

```
#include<stdio.h>
int main()
{
    int radius;
    // Khi ban khai bao 1 bien, ma chua intialize (khoi tao gia tri) cho no,
    // Kha nang 1: code van  chay duoc, nhung ra 1 gia tri rac
    // Kha nang 2: compile / build bi loi (mau do)
    // Kha nang 3: ngon ngu lap trinh, da gan gia tri mac dinh, vd 0

    // Loi khuyen: hay dam bao bien cua ban duoc gan gia tri mac dinh / nhap,
    // truoc khi su dung de tinh toan hoac de in ra man hinh
    printf("Nhap ban kinh: ");
    scanf("%d", &radius);
    printf("Dien tich hinh tron: %lf\n", 3.14 * radius * radius);
    return 0;
}
```

- Markdown
- Topic 1
- Topic 2

Quá trình từ cpp ra exe

Source code C++: .cpp, .h

Build / Compile / Biên dịch: file .cpp ⇒ file .o ⇒ file .exe

Start Debugging / Start Without Debugging / Run: chạy file .exe

Có một số lỗi có khả năng xảy ra trong quá trình từ file .cpp sang file .exe

- Viết mã nguồn, sai cú pháp, lỗi compiled-time error ⇒ ko tạo file .o
- Build → Run lần 1 → Sửa code → Build → Lần 2 bị lỗi

Build lần 1 → ko có lỗi cú pháp → tạo ra file .exe → version 1 → Build lần 2 → Báo có sai cú pháp → Có muốn Run → Yes → Chạy lên → Chạy version củ (version 1) chương trình mình đang code.

```
printf("Nhap a: ");
scanf("%d", &a);
printf("Nhap b: ");
scanf("%d", &b);
```

It's not a bug. It's a feature

Máy tính ko bao giờ có lỗi → Chỉ có lập trình viên có lỗi

Khách hàng ko bao giờ có lỗi → Chỉ có lập trình viên hiểu sai, hiểu thiếu vấn đề

- Khi bạn đọc 1 đề bài, phần nào bạn ko hiểu, hiểu ko rõ ràng → hỏi lại
- Khi đi thi, phần nào bạn ko hiểu, mập mờ quá → hỏi giám thị → bạn ghi giả sử / giả thiết / assumption ở trong bài làm
- Khi gặp tiếp xúc khách hàng, lắng nghe, hỏi lại, xác nhận lại

Copy - Paste - Edit

Edit thiếu

2 lí do để viết và sử dụng hàm

- Duplicate code → ko viết hàm
  - copy - paste - edit → edit thiếu
  - sửa yêu cầu → sửa lại ở tất cả những chỗ duplicate
- Có những cái hàm, viết 1 lần, sử dụng 1 lần, nhưng vẫn viết:
  - Lo xa → sau này, sử dụng lần 2, lần 3...
  - Làm cho hàm main ngắn lại → source code dễ đọc hơn

Thuật ngữ

3 tên gọi:

- Hàm / function
- Chương trình con / sub program
- Thủ tục / procedure

3 giai đoạn:

- Khai báo hàm / Function declaration / Function prototype: làm 1 lần
  - Tên hàm / Function name
  - Đầu vào / Tham số / Parameter / Argument
  - Đầu ra / Trả về / Return type
- Định nghĩa hàm / Cài đặt hàm / Function definition / Function implementation: làm 1 lần
  - Tên hàm / Function name
  - Đầu vào / Tham số / Parameter / Argument
  - Đầu ra / Trả về / Return type
  - Ruột / Body → có những dòng return
- Gọi hàm / Sử dụng hàm / Call a function / Invoke a function: làm nhiều lần (sử dụng nhiều lần)

Cách thực hành viết hàm

3 bước:

- Xác định phần Input, phần Process, phần Output trong chương trình của mình
- Chuyển đổi phần Process thành 1 hàm
- Thay thế đoạn Process trong hàm main thành 1 dòng code gọi sử dụng hàm

Bài 51

```
int foo(int x);
```