

BÀI TẬP THỰC HÀNH TUẦN 1

Nội dung:

- Ôn tập về C++
- Cài đặt Class đơn giản

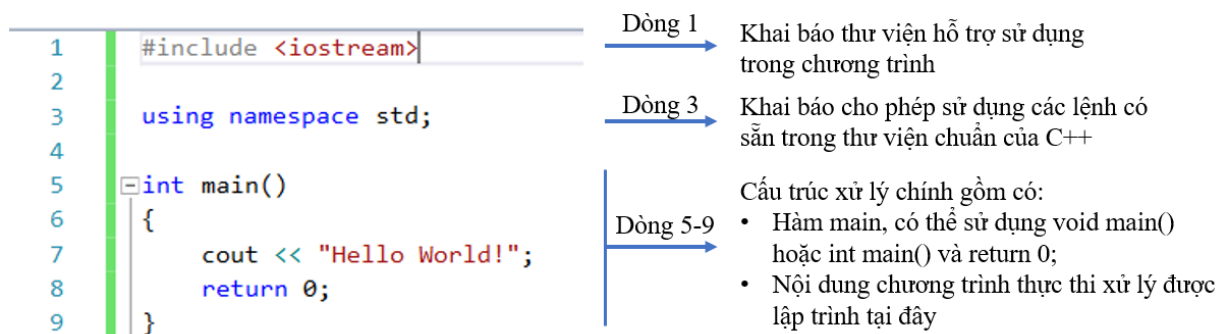
Qui định nộp bài tập:

- Mỗi bài tập tương ứng với 1 project, tên Project là BaiXX, với XX là thứ tự của bài tập (Ví dụ bài tập 1 tên Project tương ứng là Bai01).
- Tất cả các bài tập được đặt bên trong một thư mục, tên thư mục theo qui định như sau: **BTH1_HoVaTen_MSSV**. Ví dụ Sinh viên Nguyễn Văn A có MSSV là 19521234 thì đặt tên thư mục như sau: **BTH1_NguyenVanA_19521234**
- Sau đó nén thư mục trên thành tập tin **.zip hoặc .rar** (tên file nén cũng theo qui định như tên thư mục). Ví dụ **BTH1_NguyenVanA_19521234.rar**
- Lưu ý xóa thư mục được phát sinh sau khi biên dịch (thư mục Debug, .vs) của mỗi project
- Hình thức nộp bài: Nộp trên website môn học theo thời gian qui định.
- **Những bài nộp không đúng qui định như trên sẽ không được chấm điểm (0 điểm)**
- **Tất cả các bài làm có tính chất sao chép (copy) sẽ nhận 0 điểm**

1. Ôn tập về C++

a. Cấu trúc của một chương trình C++

- Cấu trúc của một chương trình C++ cơ bản gồm các phần theo thứ tự sau:
 - a) Phần khai báo thư viện nằm ở đầu chương trình với các câu lệnh `#include`
 - b) Phần khai báo cho phép sử dụng các lệnh có sẵn trong thư viện chuẩn của C++ để tiện sử dụng khi lập trình
 - c) Các hàm con xử lý
 - d) Hàm main nằm ở dưới cùng để xử lý thực thi chương trình



- Ta có thể khai báo thư viện với cú pháp:

`#include <tên thư viện>`

- Một số thư viện thường hay sử dụng:
 - `#include <iostream>` : sử dụng cho nhập xuất ra màn hình console
 - `#include <math>` : sử dụng cho việc xử lý tính toán số học
 - `#include <string>` : sử dụng cho việc xử lý chuỗi
 - `#include <vector>` : sử dụng cho việc lưu trữ dữ liệu dạng vector
 - `#include <ctime>` : sử dụng cho việc xử lý thời gian
- Một số lưu ý:
 - Nên theo **đúng thứ tự** cấu trúc xử lý chương trình, đặc biệt là các hàm con phải nằm trước hàm main, do chương trình thực thi tuần tự từ trên xuống dưới. Nếu hàm main được gọi trước các hàm con thì các hàm con sử dụng trong hàm main sẽ không được tìm thấy và chương trình báo lỗi.
 - Các lệnh xử lý trong chương trình đa phần nằm ở tập tin `.cpp`, ngoài ra còn có tập tin `.h` dùng để khai báo trước các hàm và lớp sẽ sử dụng. Do đó có thể sử dụng tập tin `.h` để khai báo trước cấu trúc của chương trình.

b. Nhập và xuất ra màn hình Console trong C++

- C++ cung cấp các lệnh có sẵn để hỗ trợ việc nhập xuất ra màn hình, dựa trên thư viện chuẩn của C++, để sử dụng, đầu chương trình cần phải khai báo ở đầu chương trình

```
#include<iostream>

using namespace std;
```

- Các lệnh nhập xuất sử dụng trong C++ :
 - cin : nhập dữ liệu, sử dụng toán tử >>
 - cout : xuất dữ liệu, sử dụng toán tử <<
- Ghi nhớ tránh nhầm lẫn : Cin đi vào biến, biến đi ra Cout (Cin >>, Cout <<)
- Cú pháp sử dụng để nhập dữ liệu vào biến :

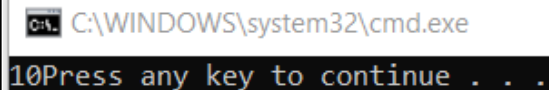
```
int variable;

cin >> variable;
```

- Màn hình sẽ chờ người dùng nhập dữ liệu vào
 - Dữ liệu sau khi nhập, nhấn Enter để hoàn tất quá trình nhập
- Cú pháp sử dụng để xuất dữ liệu ra màn hình console :

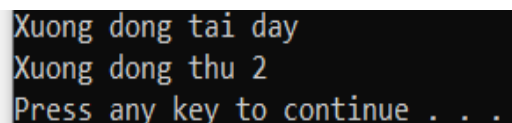
```
int variable = 10;

cout << variable;
```



- Các ký tự đặc biệt :
 - Ký tự xuống dòng : “\n” hoặc endl
 - Ký tự cách dòng tab : “\t”
 - Ký tự kết thúc chuỗi : “\0”

```
cout << "Xuong dong tai day" << endl;
cout < "Xuong dong thu 2\n";
```



2. Lập trình hàm và lập trình hướng đối tượng

a. Lập trình hàm – lập trình hướng thủ tục

- Một chương trình khi xử lý được chia thành nhiều hàm, trong đó mỗi hàm xử lý có thể có hàm con
- Mỗi hàm đều có tên hàm, tham số đầu vào hàm và kết quả trả về
- Đầu vào có thể truyền theo dạng tham trị hoặc tham chiếu
 - Truyền tham trị : giá trị biến sau khi ra khỏi hàm sẽ không thay đổi
 - Truyền tham chiếu : giá trị biến sau khi ra khỏi hàm có thể bị thay đổi

```
//Hàm cộng hai số nguyên
//Biến a truyền tham trị, biến b truyền tham chiếu
int Cong2SoNguyen(int a, int &b)
{
    .
    .
}
```

- Đặc điểm của lập trình theo hướng thủ tục :
 - Dựa theo chức năng làm trọng tâm : mỗi hàm xử lý thực hiện một số hành động hay thao tác nào đó.
 - Chương trình gọi các hàm xử lý tuần tự, truyền dữ liệu vào và trả về kết quả

b. Lập trình hướng đối tượng

- Thay vì tập trung vào xử lý dựa theo chức năng làm trọng tâm, lập trình hướng đối tượng tập trung vào đối tượng cần xử lý, trong đối tượng đó sẽ có khả năng thực hiện chức năng gì và có những thao tác, thuộc tính nào có liên quan
- Mỗi lớp đối tượng được dùng để mô tả một nhóm các đối tượng cùng loại có cùng một cách ứng xử trong thế giới thực.
- Các thành phần trong lớp đối tượng (**class**) :
 - Tên lớp : mỗi lớp sẽ có tên duy nhất phân biệt với lớp khác trong cùng phạm vi
 - Thành phần dữ liệu hay thuộc tính (**attributes**): các trường để lưu dữ liệu của lớp đối tượng
 - Các phương thức của lớp (**methods**): là các hàm xử lý viết riêng cho đối tượng của lớp.
- Cài đặt lớp đối tượng, cần phải xác định các thành phần trong lớp đối tượng
- Mỗi thuộc tính và phương thức sẽ có **tầm vực truy cập (scope)**
 - **Public** : cho phép truy xuất thuộc tính và phương thức từ bên ngoài lớp
 - **Private** : chỉ cho phép truy xuất, thay đổi thuộc tính, phương thức ở bên trong nội bộ của lớp đó
 - Nếu không khai báo tầm vực, **mặc định sẽ là private.**

```

#include<iostream>

using namespace std;

//Tên class: PhanSo
class PhanSo
{

//Phần khai báo thuộc tính
//Nếu để mặc định thì sẽ là private
    int tuso;
    int mauso;

//Phần khai báo phương thức bên trong lớp đối tượng
public:
    void NhapPhanSo()
    {
        cout << "Moi nhap tu so: ";
        cin >> tuso;
        cout << "Moi nhap mau so: ";
        cin >> mauso;
    }

    //Khai báo xử lý phương thức bên ngoài đối tượng
    void XuatPhanSo();
};

```

Ví dụ về khai báo lớp đối tượng Phân số gồm có tử số và mẫu số

- Có 2 cách để khai báo và xử lý phương thức của lớp đối tượng :

- Khai báo và xử lý trực tiếp bên trong đối tượng đó
- Khai báo trước bên trong đối tượng, bên ngoài xử lý riêng
- Sử dụng đối tượng : khai báo trước thông tin đối tượng, gọi các phương thức xử lý tương ứng (xem ví dụ trong mã nguồn phía trên)

3. Bài tập thực hành

Bài tập 1: Viết chương trình C++ cài đặt đối tượng phân số, trong đó phân số được định nghĩa gồm có tử số và mẫu số với kiểu dữ liệu là số nguyên. Hãy cài đặt các thao tác trên đối tượng phân số này:

- Nhập phân số
- Xuất phân số vừa nhập theo 2 dạng:
 - Xuất theo định dạng tử số / mẫu số
 - Xuất theo định dạng kết quả phép chia của tử số và mẫu số (số thập phân)
- Nhập vào 2 phân số, xuất ra kết quả **cộng, trừ, nhân, chia** của 2 phân số vừa nhập

Bài tập 2: Viết chương trình C++ cài đặt đối tượng số phức, trong đó số phức được định nghĩa gồm có phần thực và phần ảo với kiểu dữ liệu là số thực. Hãy cài đặt các thao tác trên đối tượng phân số này:

- Nhập số phức
- Xuất số phức vừa nhập theo dạng: $a + bi$ (Với a, b lần lượt là phần thực và phần ảo)
- Nhập vào 2 số phức, xuất ra kết quả **cộng, trừ, nhân, chia** của 2 số phức vừa nhập.

Ví dụ: Cho hai số phức $A(a_1, a_2), B(b_1, b_2)$

- $A + B = (a_1 + b_1, a_2 + b_2)$
- $A - B = (a_1 - b_1, a_2 - b_2)$
- $A * B = (a_1 * b_1 - a_2 * b_2, a_1 * b_2 + a_2 * b_1)$
- $A / B = \left(\frac{a_1 * b_1 + a_2 * b_2}{b_1^2 + b_2^2}, \frac{b_1 * a_2 - a_1 * b_2}{b_1^2 + b_2^2} \right)$

Bài tập 3: Viết chương trình C++ cài đặt đối tượng điểm trong không gian Oxy, trong đó một điểm được định nghĩa gồm có tọa độ x và tọa độ y đều là số thực. Hãy cài đặt các thao tác trên đối tượng điểm này:

- Nhập một điểm và xuất điểm vừa nhập ra màn hình
- Nhập 2 điểm và xuất ra khoảng cách giữa 2 điểm vừa nhập trong không gian Oxy

Bài tập 4: Xây dựng lớp Candidate (Thí sinh) gồm các thuộc tính: mã, tên, ngày tháng năm sinh, điểm thi Toán, Văn, Anh và các phương thức cần thiết.

Xây dựng lớp TestCandidate để kiểm tra lớp trên:

- Nhập vào n thí sinh (n do người dùng nhập) và xuất ra màn hình thông tin của từng thí sinh (mã, tên, ngày tháng năm sinh, điểm thi Toán, Văn, Anh, điểm trung bình Toán, Văn, Anh)
- In ra thông tin về các thí sinh có tổng điểm lớn hơn 15