**Lab 04**

**Hàm tạo sao chép**

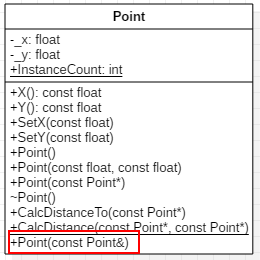
# Lập trình hướng đối tượng

|  |  |
| --- | --- |
| **Mục tiêu** | Cài đặt và sử dụng hàm tạo sao chép |

**Bộ môn Công nghệ phần mềm – Khoa Công nghệ thông tin**



# Hướng dẫn khởi đầu

** Mô tả bài tập**

Cho trước thiết kế lớp **Điểm** trong không gian hai chiều với 2 thuộc tính **x** và **y.**

Hãy cài đặt cụ thể lớp này với các thành phần:

+ Thuộc tính private

+ Các hàm getter setter tương ứng.

+ Hàm tạo và hàm hủy

+ Hàm tạo có đối số

+ Hàm CalcDistanceTo để tính khoảng cách đến điểm khác

+Thành phần tĩnh InstanceCount đếm số lượng thể hiện đã tạo ra của lớp Điểm

+ Hàm tĩnh CalcDistance để tính khoảng cách giữa hai điểm

**+ Hàm tạo sao chép để khởi tạo thông tin từ một điểm khác**

**Hướng dẫn cài đặt**

**Bước 1: Tạo mới dự án**

- Chọn loại dự án là **C++** / **Console Application**.

- Đặt tên solution là: **CopyConstructor**. Đặt tên project là **PointV4**

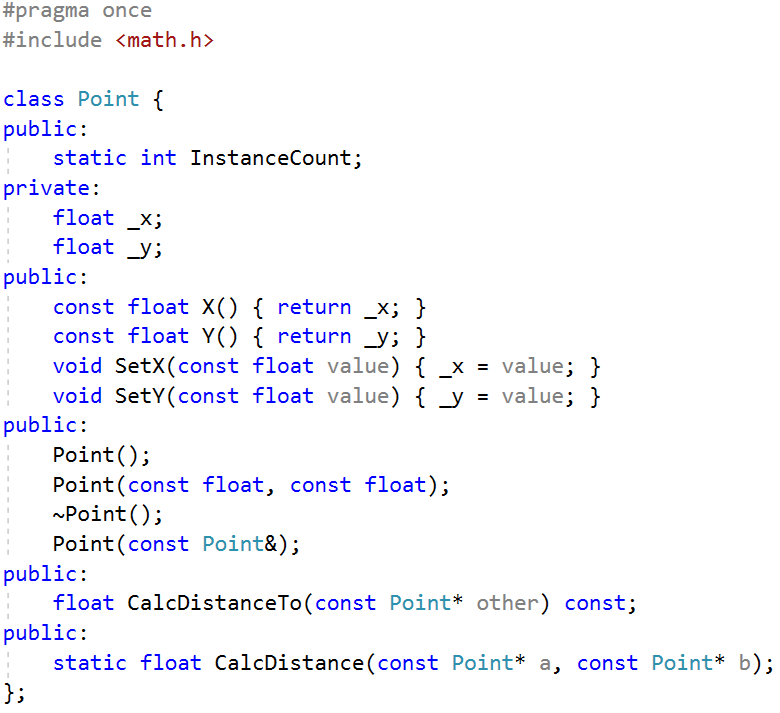
- Nếu sử dụng Visual Studio 2017 trở lên cần vô hiệu hóa **Precompiled header** bằng cách nhấn phải vào project chọn Properties. Vào mục **C / C++** > **All Options**, tìm tới tùy chọn **Precompiled header** và chọn **Not using precompiled headers**.

**Bước 2: Tạo định nghĩa lớp trong file Point.h (CPoint.h)**

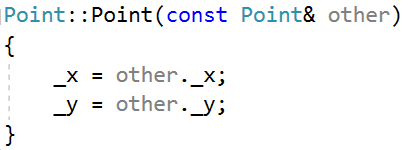
- Thêm một tập tin header bằng cách nhấn phải vào project, chọn **Add > New Item**…

- Chọn loại tập tin là **Header File (.h),** đặt tên là Point.h

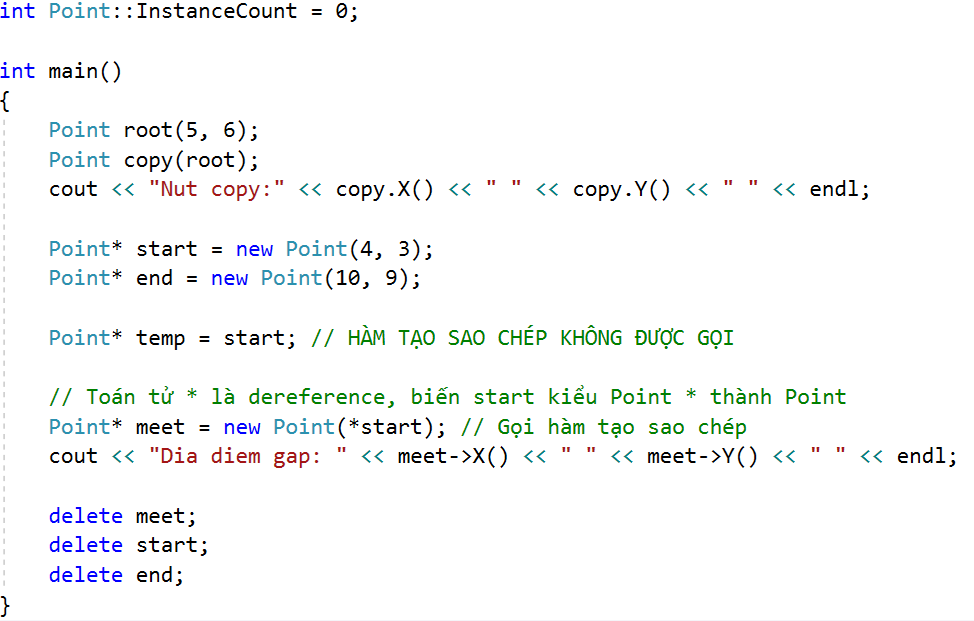
- Tạo ra định nghĩa lớp như sau:



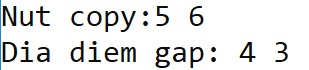
**Bước 3: Cài đặt thêm hàm tạo sao chép trong file Point.cpp**



**Bước 4: Cài đặt hàm main để test việc cài đặt của lớp Point (CPoint)**



Chạy lên và thấy kết quả như sau:



# Bài tập vận dụng

**Yêu cầu**

1. Thực hiện định nghĩa lớp theo thiết kế cho trước vào tập tin .h.
2. Thực hiện cài đặt lớp trong tập tin .cpp cho lớp tương ứng.
3. Viết các đoạn mã nguồn kiểm tra việc định nghĩa lớp trong hàm main.

**Danh sách bài tập cụ thể**

1. Lớp **Đường thẳng** có hai thành phần **Điểm**: **Bắt đầu** và **Kết thúc**.

Gợi ý: Tên project: **LineV3**

+ Tên lớp: Line / CLine

+ Thành phần: \_start, \_end

+ Thuộc tính: Length cho biết độ dài của đường thẳng

+ Thuộc tính **tĩnh** *InstanceCount* cho biết đã tạo ra bao nhiêu đối tượng từ lớp đường thẳng.

+ Hàm tạo sao chép để tạo ra đường thẳng từ một đường thẳng khác

2. Lớp **Hình chữ nhật** có hai thành phần **Điểm**: **Trái trên** và **Phải Dưới**

Gợi ý: Tên project: **RectangleV3**

+ Tên lớp: Rectangle / CRectangle

+ Thành phần: \_topLeft, \_bottomRight

+ Thuộc tính **tĩnh** *InstanceCount* cho biết đã tạo ra bao nhiêu đối tượng từ lớp hình chữ nhật

+ Hàm tạo sao chép để tạo ra hình chữ nhật từ một hình chữ nhật khác

3. Lớp **Mảng động** (**DynamicArray**) có 3 thành phần

+ int\* \_a: chứa mảng các số nguyên

+ int \_len: chứa độ dài hiện tại của mảng

+ int \_max: chứa độ dài tối đa mảng hỗ trợ

Khi khởi tạo, tạo mảng mặc định có **128** phần tử.

Khi hủy, nhớ thu hồi vùng nhớ cấp phát cho mảng

Thành phần cơ bản gồm:

+ **PushBack**(int value) : thêm 1 phần tử vào mảng. Cài đặt hiện tại không cần tính tới việc vượt quá khả năng của mảng. Ta sẽ cài ở tuần sau. Cứ việc đẩy phần tử mới vào mảng ở phiên bản này.

+ **GetAt**(int i): Lấy một phần tử tại vị trí i. Cài đặt hiện tại không cần tính tới việc truy cập phần tử không hợp lệ. Ta sẽ cài đặt ở tuần sau. Cứ việc trả ra phần tử tại vị trí thứ i.

+ Hàm tạo sao chép để tạo ra một mảng động khác từ một mảng động trước đó.

# Hướng dẫn nộp bài

**Trước khi nộp cần chú ý:**

- Lấy tập tin exe được biên dịch sẵn trong thư mục Debug, copy nó ra thư mục Release bên ngoài mã nguồn.

- Xóa hết tất cả các tập tin trung gian trong quá trình biên dịch bằng cách chọn **Build** > **Clean**.

- Chú ý thư mục ẩn **.vs** rất nặng. Cần hiển thị file ẩn mới thấy và xóa nó đi được.

Nếu bạn muốn biết cách làm đúng thì cần tự tìm cách build ở chế độ Release và copy file exe kết quả ra bên ngoài để nộp mới đúng. Tuy nhiên nếu chưa hiểu ý nghĩa thì cứ lấy đại file exe có sẵn đi nộp cũng được (hiện tại đang trong thư mục Debug ứng với chế độ biên dịch Debug)

**Tổ chức bài nộp**

+ Thư mục **Source**: chứa mã nguồn đã được clean

+ Thư mục **Release**: chứa tập tin thực thi đã được biên dịch từ mã nguồn

+ Tập tin **readme.txt**: chứa thông tin sinh viên, gồm MSSV và họ tên. Ghi chú kèm các thông tin giáo viên cần chú ý khi chấm bài.

Để nộp bài, nén tất cả lại và đặt tên với định dạng **MSSV.zip** hoặc **MSSV.rar** và nộp.

**Nếu làm đúng các bước trên file này sẽ có kích thước < 100 KB!**

(Tuy nhiên cũng đừng quá lo lắng nếu nó khác con số trên, miễn < 12 MB để nộp được trên moodle là okie nhé!)

**Ngoài lề**: Để đảm bảo sau này nhìn vào file nén còn biết ngay nó làm gì, ta nên thêm vào một số thông tin theo sau MSSV. Ví dụ: 0712221-Lab04-CopyConstructor.zip. Tuy nhiên việc này là KHÔNG bắt buộc nhé.

**-- HẾT --**