**Lab 06**

**Nhập xuất**

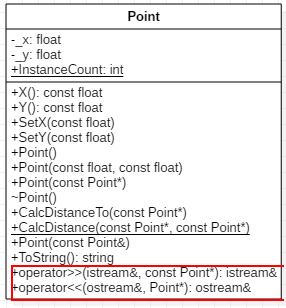
# Lập trình hướng đối tượng

|  |  |
| --- | --- |
| **Mục tiêu** | 1. Nhập các thông tin của đối tượng từ bàn phím 2. Xuất đối tượng với đầy đủ thông tin ra màn hình |

**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

+ Hàm ToString() để chuyển một đối tượng thành chuỗi và xuất ra màn hình

**+ Nạp chồng toán tử nhập (operator>>) để nhập liệu tạo ra đối tượng**

**+ Nạp chồng toán tử xuất (operator<<) để xuất đối tượng ra màn hình**

**Hướng dẫn cài đặt**

**Bước 1: Tạo mới dự án**

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

- Đặt tên solution là: **InputOutput**. Đặt tên project là **PointV6**

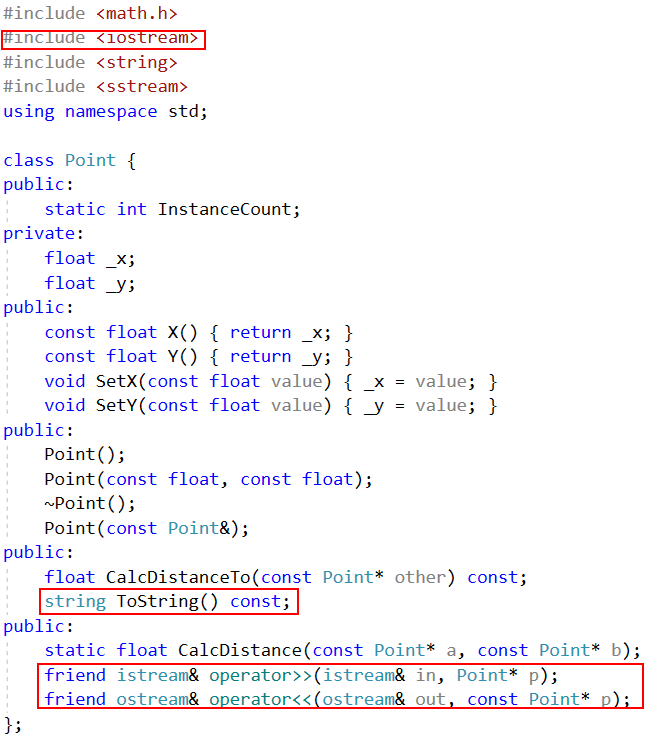
- Nếu sử dụng Visual Studio 2017 trở lên cần vô hiệu hóa **Precompiled header** bằng cách nhấn phải vào project chọn Properties. Vào mục **C / C++** > **All Options**, tìm tới tùy chọn **Precompiled header** và chọn **Not using precompiled headers**.

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

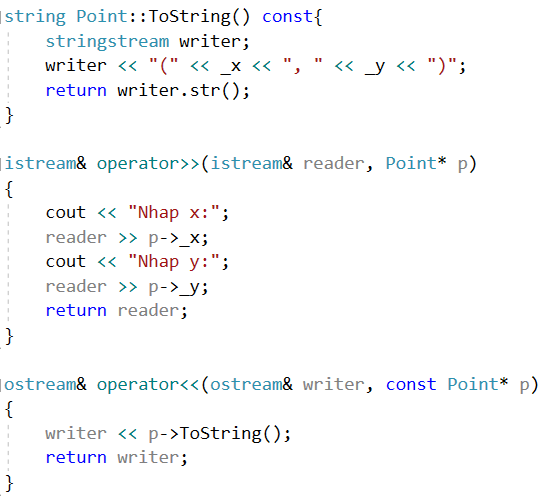
- Thêm một tập tin header bằng cách nhấn phải vào project, chọn **Add > New Item**…

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

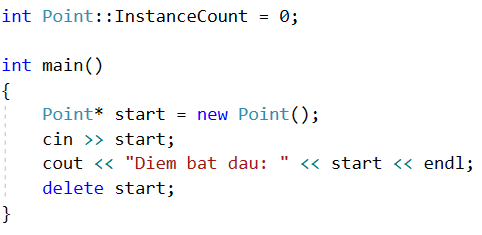
- Tạo ra định nghĩa lớp như sau:

****

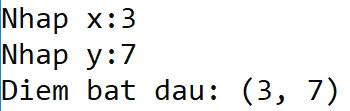
**Bước 3: Cài đặt thêm hàm nhập và xuất trong file Point.cpp**



**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.

+ Bổ sung cài đặt toán tử nhập (operator>>)

+ Bổ sung cài đặt toán tử xuất (operator<<)

1. Viết các đoạn mã nguồn kiểm tra việc định nghĩa lớp trong hàm main.

**Danh sách các lớp cần cải tiến cụ thể**

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

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

3. Lớp **Hình tam giác** có ba thành phần **Điểm** ứng với 3 đỉnh : **a, b, c**.

4. Lớp **Hình tròn** có 2 thành phần: **tâm** (Lớp **Điểm**) và **bán kính** (số thực).

5. Lớp **Phân số** có 2 thành phần: **tử** (số nguyên) và **mẫu** (số nguyên)

6. Lớp **Sinh viên** có 3 thành phần: **họ** (chuỗi), **tên lót** (chuỗi) và **tên** (chuỗi).

7. Lớp **Mảng động** (**DynamicArray**)

8. Lớp **Thời gian** (**Time**) với 3 thành phần **giờ** (số nguyên), **phút** (số nguyên), **giây** (số nguyên)

+ Hàm tĩnh: **IsValid**(int, int, int) cho biết 3 biến truyền vào có tạo thành thời gian hợp lệ hay không

9. Lớp **Ngày Tháng** (**Date**) với 3 thành phần **ngày** (số nguyên), **tháng** (số nguyên), **năm** (số nguyên)

+ Hàm tạo không đối: tự lấy ngày hiện tại để khởi tạo

+ Hàm tạo có đối: khởi tạo từ ngày tháng năm truyền vào.

+ Hàm tĩnh: **IsLeapYear**(int) cho biết năm có phải năm nhuận hay không

+ Hàm tĩnh: **IsValid**(int, int, int) cho biết 3 biến ngày tháng năm truyền vào có tạo thành ngày tháng hợp lệ không

# 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-Lab06-InputOutput.zip. Tuy nhiên việc này là KHÔNG bắt buộc nhé.

**-- HẾT --**