**Lab 02**

**Hành vi của lớp**

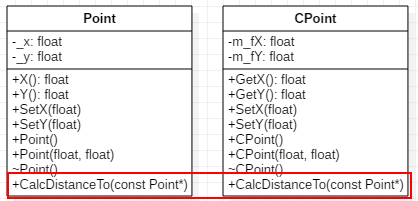
# Lập trình hướng đối tượng

|  |  |
| --- | --- |
| **Mục tiêu** | 1. Cài đặt hành vi của đối tượng 2. Cài đặt getter phái sinh 3. Cấp phát và thu hồi vùng nhớ cho thành phần |

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



# Hướng dẫn khởi đầu 1 – Hành vi

**Mô tả bài tập**

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

Cài đặt hàm tính khoảng cách từ một điểm đến một điểm khác

Thiết kế tinh gọn và theo kí pháp Hungary

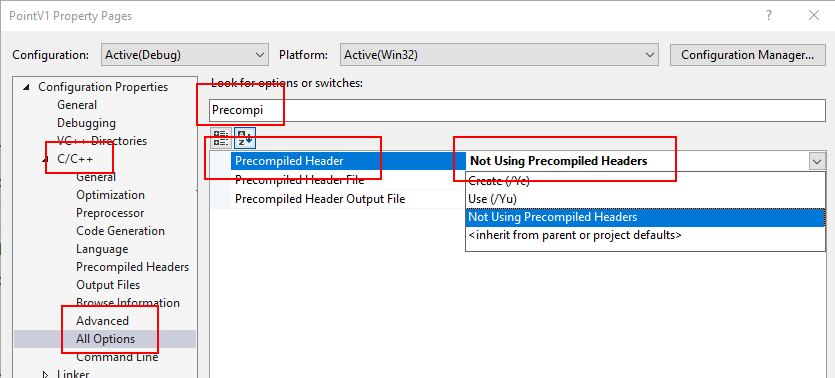
**Hướng dẫn cài đặt**

**Bước 1: Tạo mới dự án**

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

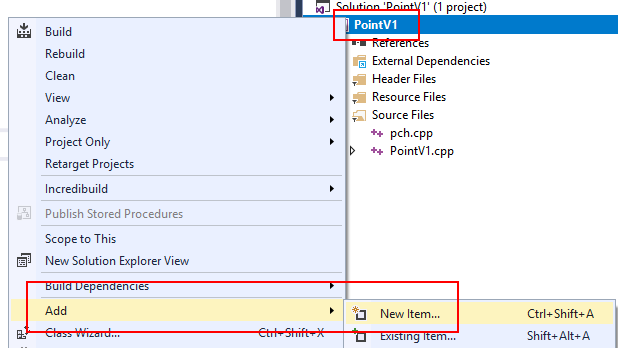
- Đặt tên solution là: **ClassWithBehaviors**. Đặt tên project là **PointV2**.

- 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**.

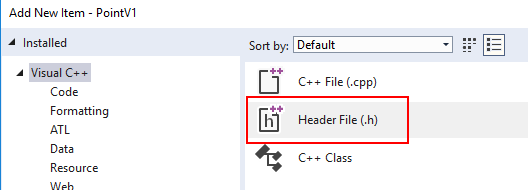


**Bước 2: Tạo định nghĩa lớp trong file Point.h (CPoint.h)**

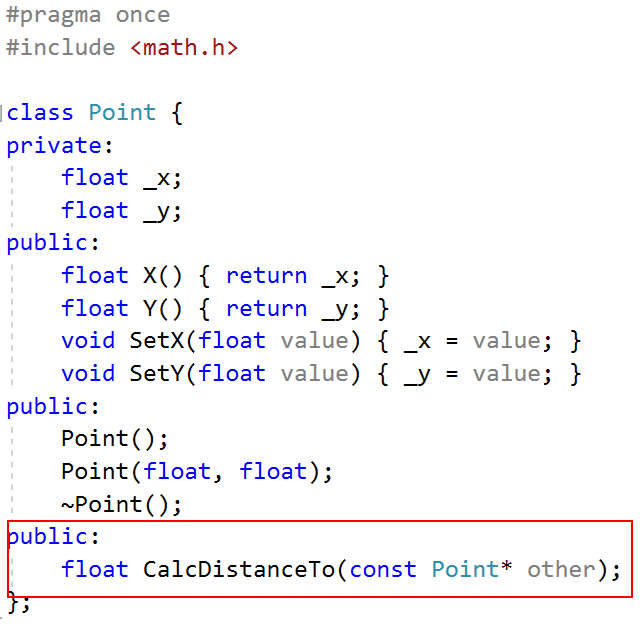
- Thêm một tập tin header bằng cách nhấn phải vào project, chọn **Add > New Item**…



- Chọn loại tập tin là **Header File (.h),** đặt tên là Point.h (*hoặc CPoint.h nếu bạn muốn sử dụng kí pháp Hungary*).

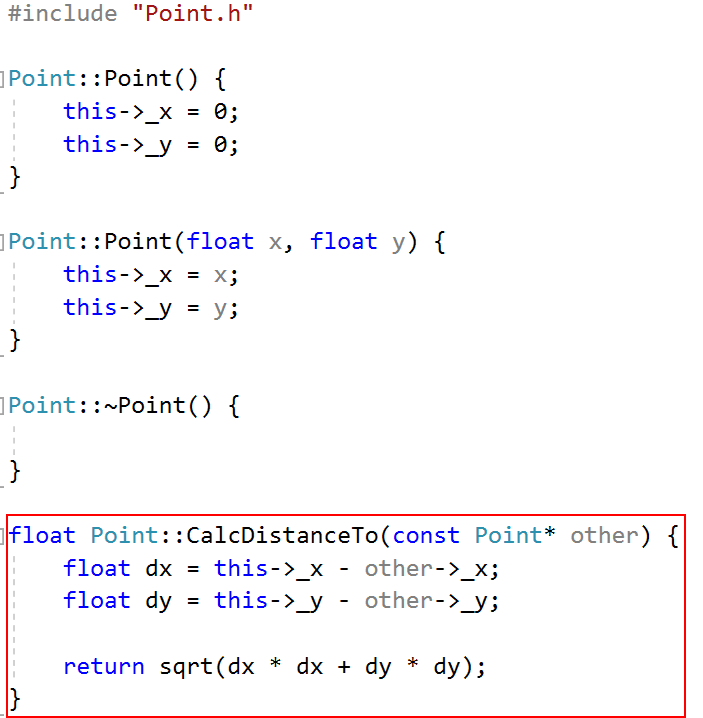


- Tạo ra định nghĩa lớp như sau:



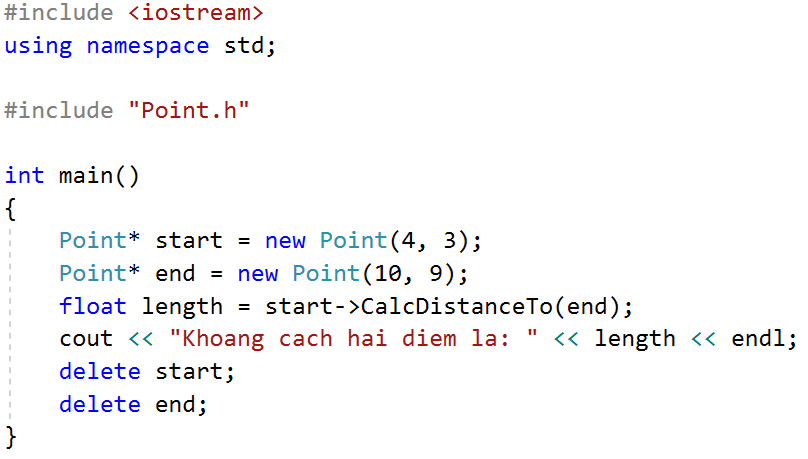
Chú ý từ khóa const ám chỉ ta không muốn thay đổi giá trị của thành phần kiểu dữ liệu Point mà con trỏ other đang trỏ đến.

**Bước 3: Cài đặt lớp trong file Point.cpp (CPoint.h)**



Chú ý bên trong hàm CalcDistanceTo có thể truy xuất thành phần private của đối tượng con trỏ other đang trỏ đến. Vì cả hai (this và other) đều trỏ đến **cùng một loại** đối tượng nên hàm bên trong lớp Point có thể truy xuất đến thuộc tính private, vốn đáng lẽ bị che dấu với các đối tượng khác kiểu.

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



Chạy lên và thấy kết quả như sau:



# Hướng dẫn khởi đầu 2 – Getter phái sinh

**Mô tả bài tập**

Cho trước thiết kế lớp **Đường tròn** trong không gian hai chiều với 2 thuộc tính **Tâm** và **Bán kính.**

Cài đặt hàm lấy các thông tin đặc trưng của đường tròn:

+ Đường kính (Diameter)

+ Chu vi (Perimeter)

+ Diện tích (Area)

**Hướng dẫn cài đặt**

**Bước 1: Thêm mới dự án**

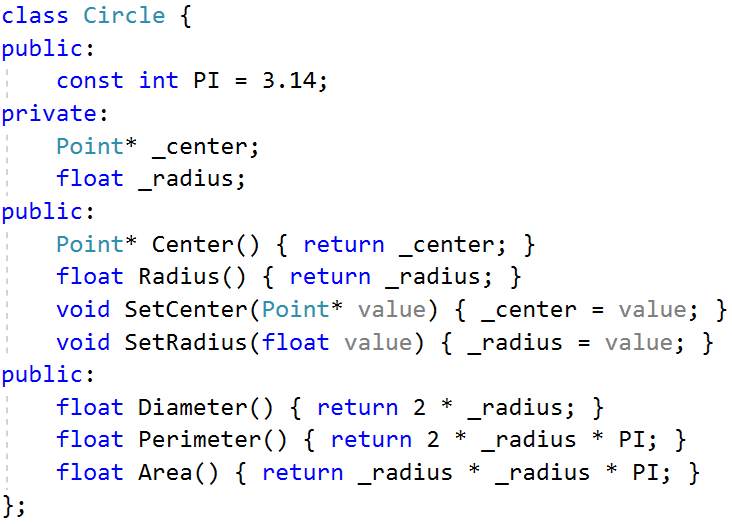
- Thêm mới một dự án vào solution và đặt tên là **CircleV2**.

**Bước 2: Tạo định nghĩa lớp trong file Circle.h**

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

- Chọn loại tập tin là Header file và đặt tên là **Circle.h**

- Nội dung khai báo lớp như bên dưới:

****

**Chú ý:** Tên hàm thường bắt đầu bằng động từ. Khi ta gặp tên hàm bắt đầu bằng danh từ thì có thể hiểu đó là hàm lấy property.

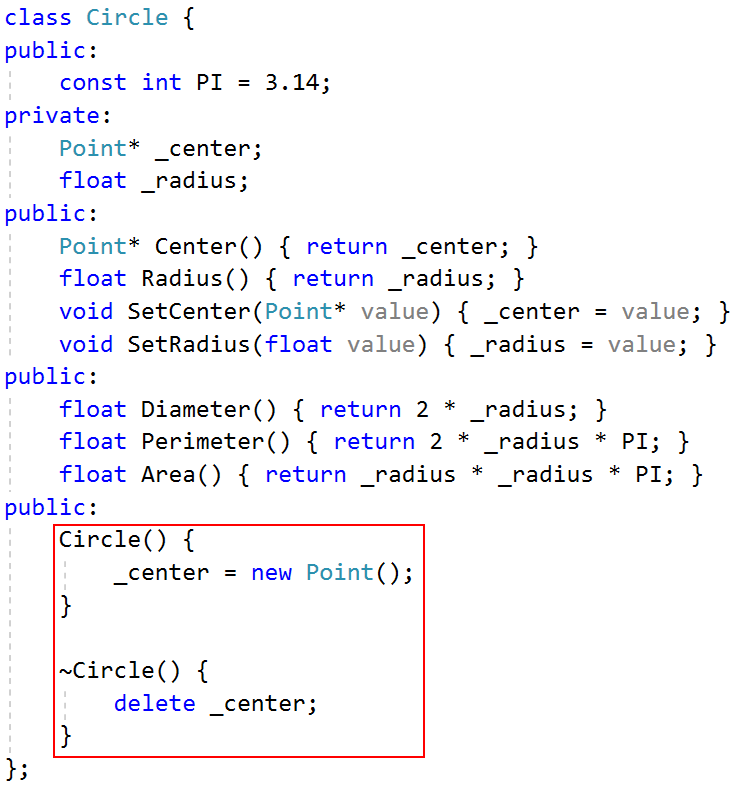
# Hướng dẫn khởi đầu 3 – Cấp phát và thu hồi

**Mô tả bài tập**

Cải tiến cài đặt lớp **Đường tròn**, trong hàm tạo cấp phát vùng nhớ cho lớp Điểm, trong hàm hủy thu hồi vùng nhớ đã cấp phát

**Hướng dẫn cài đặt**

Bổ sung thêm hai cài đặt như sau:



**Chú ý:** Cài đặt đúng cần tách ra file .h và .cpp riêng! Tự viết hàm main kiểm tra lớp!

# 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.
   1. Thuộc tính có đầy đủ getter / setter
   2. Có hàm tạo, hàm hủy thực hiện cấp phát và thu hồi các thành phần
3. Viết các đoạn mã nguồn kiểm tra việc định nghĩa lớp trong hàm main.

**Danh sách bài tập cụ thể**

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

Gợi ý: Tên project: RectangleV2

+ Tên lớp: Rectangle / CRectangle

+ Thành phần: \_topLeft, \_bottomRight

+ Chu vi, diện tích

+ Hàm tạo mặc định khởi tạo 2 điểm ở tọa độ (1, 1) và (10, 10).

+ Hàm hủy: thu hồi vùng nhớ của \_topLeft và \_bottomRight

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

Gợi ý: Tên project: TriangleV2

+ Tên lớp: Triangle / CTriangle

+ Thành phần: \_a, \_b, \_c

+ Chu vi, diện tích

+ Hàm tạo mặc định khởi tạo 3 điểm (1, 1), (1, 2), (4, 2)

+ Hàm hủy thu hồi vùng nhớ 3 điểm thành phần a, b, c.

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

Gợi ý: Tên project StudentV2

+ Tên lớp: Student / CStudent

+ Thành phần: \_firstName, \_middleName, \_lastName. Kiểu dữ liệu: string. Cần include <string>

+ Họ tên (đầy đủ): gợi ý tên Fullname. Tạo ra từ 3 thành phần first, middle, last. Chỉ cần sử dụng toán tử + của lớp tring là đủ

Return \_firstName + “ ” + \_middleName + “ “ + \_lastName

+ Hàm hủy: không cần làm gì

4. 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

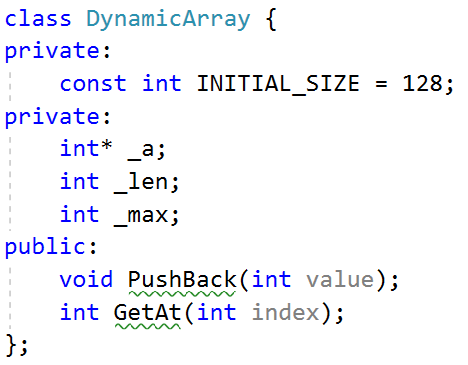
Khi khởi tạo, tạo mảng mặc định có 128 phần tử.

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.



# 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-Lab02-ClassWithBehaviors.zip. Tuy nhiên việc này là KHÔNG bắt buộc nhé.

**-- HẾT --**