

作業 #07

■#練習：請定義一個 Point 的類別，目標是找出距離最遠的兩點。

■Requirements：

1. 定義一個 Point 類別包含私有變數 x 和 y 座標，利用 Vector 存放多個點
2. 利用「-」運算子多載成計算兩點的距離
3. 利用 static 記錄所有點中距離最遠的兩個點與距離
4. 最後利用 show(...) 印出最遠的兩個點與距離

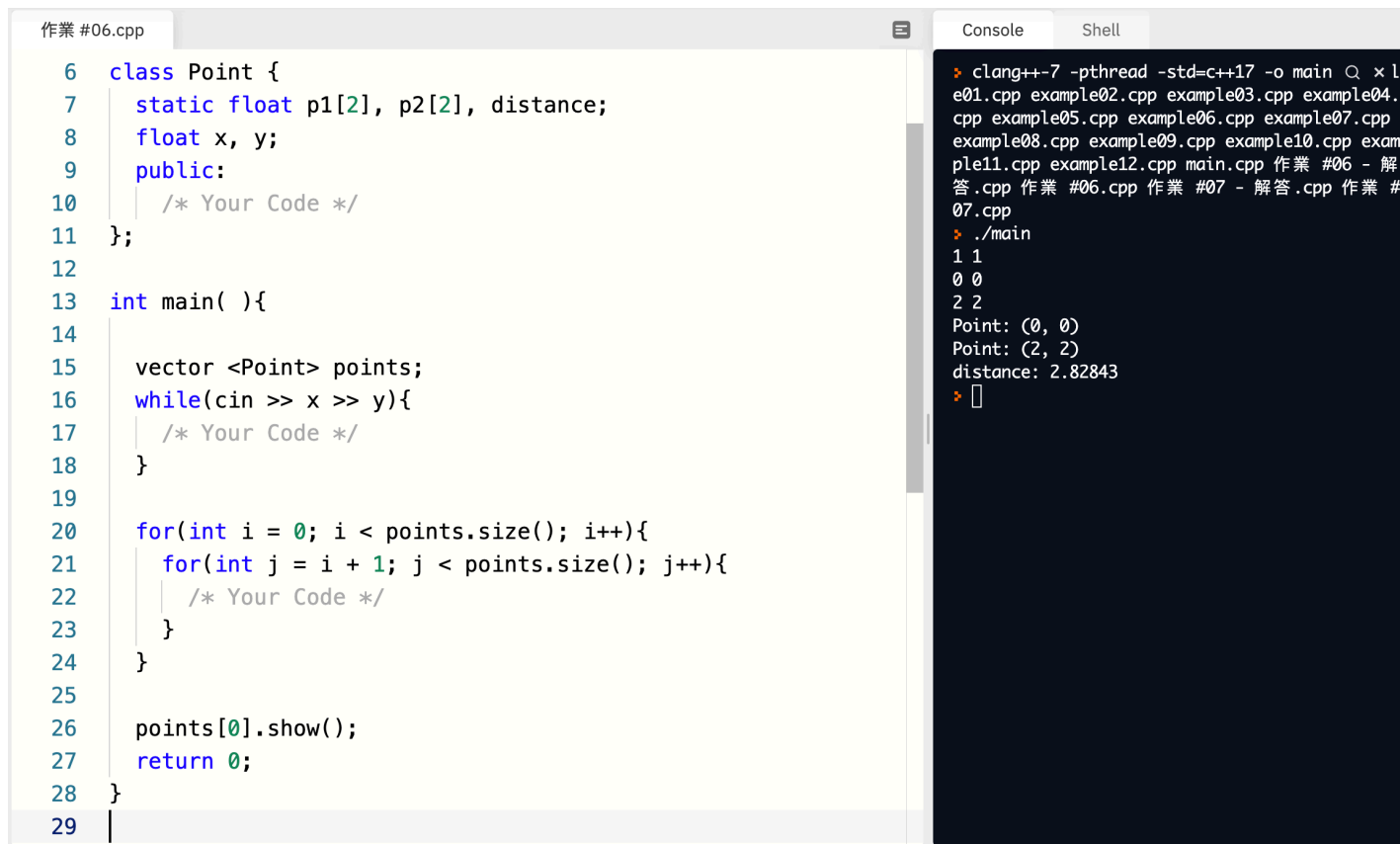
■Sample Input：參考下頁

■Sample Output：參考下頁

■Note：僅限 03/09 或 03/16 上課繳交

參考程式碼與結果

■你可以參考下列程式碼修改，也可以自己從頭開始寫。只要執行結果符合題目要求即可。



The screenshot displays a C++ development environment with two panels. The left panel, titled '作業 #06.cpp', contains the following code:

```
6 class Point {
7     static float p1[2], p2[2], distance;
8     float x, y;
9     public:
10    /* Your Code */
11 };
12
13 int main( ){
14
15     vector <Point> points;
16     while(cin >> x >> y){
17         /* Your Code */
18     }
19
20     for(int i = 0; i < points.size(); i++){
21         for(int j = i + 1; j < points.size(); j++){
22             /* Your Code */
23         }
24     }
25
26     points[0].show();
27     return 0;
28 }
29
```

The right panel, titled 'Console', shows the compilation and execution process:

```
> clang++-7 -pthread -std=c++17 -o main 0 x l
e01.cpp example02.cpp example03.cpp example04.
cpp example05.cpp example06.cpp example07.cpp
example08.cpp example09.cpp example10.cpp exam
ple11.cpp example12.cpp main.cpp 作業 #06 - 解
答.cpp 作業 #06.cpp 作業 #07 - 解答.cpp 作業 #
07.cpp
> ./main
1 1
0 0
2 2
Point: (0, 0)
Point: (2, 2)
distance: 2.82843
>
```

作業繳交說明

你需要繳交以下檔案到 **Portal** 作業：

- 1. cpp 程式碼
- 2. 程式碼內有文字說明的註解
- 3. 執行結果截圖

(若無法上傳多個檔案，請壓縮成 zip 或 rar 格式，
並且命名成「學號.zip」或「學號.rar」)

■ 本次作業截止時間：

僅限 03/09 或 03/16 上課繳交

```
main.cpp
1 // 學號： 1091000 姓名：王大明
2 /* 文字說明：這個作業當中，我利用一個變數 x 存放名字並且利用
   cout 做印出。 */
3
4 #include <iostream>
5 #include <string>
6 using namespace std;
7
8 int main() {
9     string x = "王大明";
10    std::cout << "Hello World, " << x << endl;
11 }
```

```
> clang++-7 -pthread -std=c++17 -o main main.cpp
> ./main
Hello World, 王大明
> 
```