

## 物件導向程式設計及應用第五次作業

Due: 2021/01/02 13:00

※注意事項：請依照課程網站內所公告之“作業檔案命名規則與規定”進行作業檔案命名以及繳交作業，未依照規定將斟酌扣分。

● 第一題：(15%)

附檔 functor.cpp 為課本的範例程式，在說明函數子(functors)的使用方法。本次作業將修改此範例，將函數子使用拉姆達函數(lambda functions)替換。

利用一個已命名的 lambda 取代 outint()函數；利用一個已命名 lambda 將函數物件 f100 取代；利用一個匿名 lambda 運算式將 TooBig<int>(200)取代。

● 第二題：(15%)

```
int main()
{
    using namespace std;
    // list of double deduced from list contents
    auto q = average_list({15.4, 10.7, 9.0});
    cout << q << endl;
    // list of int deduced from list contents
    cout << average_list({20, 30, 19, 17, 45, 38}) << endl;
    // forced list of double
    auto ad = average_list<double>({'A', 70, 65.33});
    cout << ad << endl;
    return 0;
}
```

請完成 average\_list()函數來完成此程式。

● 第三題：(70%)

附檔(HW5-3.bin)為一個 Binary 格式的三維點資料檔案，但其中有許多重複的點，請撰寫程式將重複的點移除、顯示點數目，並以註解的方式標示出濾完重複點的陣列。將附檔 HW5-3.cpp 加入你的專案中並續寫，將作業二的 Point 類別擴增為三維點並加入程式中，你可以另外加入自定義類別或是函數，但請勿修改附檔。

本題需撰寫兩次，第一次請不要使用任何的 C++內建函式庫，單純以多重 for 迴圈過濾重複點並計算個數(15%)。第二次則可使用任何的函式庫，設法提升你的演算效率(25%)。

請撰寫一份報告(30%)，解說你的兩種方法，並顯示結果及討論兩者的效率，探討效率提升的原因。

函數 Point\* LoadPoint(const char\*, unsigned int&); 為讀取附檔的函數，函數會回傳點陣列。

輸出點資料的格式如下：

主程式如下：

```
int main()
{
    std::chrono::steady_clock::time_point start, end;//紀錄時間的變數

    start = std::chrono::steady_clock::now();//紀錄讀檔開始時間
    unsigned int nPoint;
    Point* point_array = LoadPoint("HW5-3.bin", nPoint);//讀點
    end = std::chrono::steady_clock::now();//紀錄讀檔結束時間
    std::cout << "Time " << std::chrono::duration_cast<std::chrono::milliseconds>(end - start).count() << "
ms" << std::endl;
    std::cout << "Number of point " << nPoint << std::endl;

    start = std::chrono::steady_clock::now();//紀錄移除重複點開始時間

    //請將移除重複點演算法撰寫於此

    end = std::chrono::steady_clock::now();//紀錄移除重複點結束時間

    std::cout << "Time " << std::chrono::duration_cast<std::chrono::milliseconds>(end - start).count() << "
ms" << std::endl;
    std::cout << /*移除重複點後的點個數*/ << std::endl;

    delete[] point_array;
    system("pause");
    return 0;
}
```

※Hint: HW5-3-1.bin 移除重複點後共有 5,276 個點。

※Hint: HW5-3-2.bin 移除重複點後共有 687,166 個點。

※Hint: HW5-3-3.bin 移除重複點後共有 24 個點。