

物件導向程式設計及應用第五次作業

Due: 2018/1/21 23:59

※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：(30%)

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：(20%)

```
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：(50%)

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，並輸出點資料成 ASCII 格式之檔案。

請將附檔 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。

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：

Number of point: (移除重複點後的點個數)

X Y Z(第一點座標)

X Y Z(第二點座標)

...依序列出

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;
```

//在此加入寫檔程式碼

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

※成績計算說明：此作業共 50%，完成移除重複點的功能佔 30%，剩餘 20%將依照程式運算速率進行給分，說明如下：

1. 運算速率在 800ms 以下，可獲得全部 20%
2. 運算速率超過 800ms 則以所有同學中最慢運算速率到 800ms 內插 0%~20%，以決定分數
EX: 所有同學中最慢運算速率為 1200ms，甲同學的運算時間為 900ms，則甲同學可獲得

$$\left(\frac{1200 - 900}{1200 - 800} \right) \times 20 = 15 \text{ 分}$$

3. 若所有同學均超過 800ms 則速率最快者獲得 20%，其餘以說明 2 之方式以最快者之速率為下界進行內插

※HW5-3.bin 移除重複點後共有 687,166 個點。