

銘傳大學 99 學年度轉學生招生考試

資訊管理學系、資訊工程學系

第四節

程式設計試題

(第 1 頁共 2 頁) (限用答案本作答)

可使用計算機 不可使用計算機

1. 本考卷主考 C++，在題目未說明環境時，以 C++ 為之。
2. 是非及填充每格 2.5 分，總分四捨五入進整數。

是非題: (對請打 O)(錯請打 X)

- ①. 函數可以透過 `return by struct` 傳回一個以上的基本型別的數值。
- ②. 函數可以透過 `call by reference` 修改傳入物件。
- ③. 函數可以使用 `call by reference` 及 `return by reference` 達到子程序的功能。
- ④. 函數使用 `return by value` 時會隱然引發回傳類別的複製建構子。
- ⑤. `int* p,q;` 左式中，`p` 與 `q` 都是整數指標。
- ⑥. 陣列宣告時，陣列名稱就是陣列開頭元素的地址。
- ⑦. 宣告陣列 `int data[10];` 後，`data[3]` 與 `*(data+3)` 同義。
- ⑧. 類別繼承時，只可以繼承一個類別。
- ⑨. 陣列索引值是由 1 起算的。
- ⑩. 運算子覆載時，可以透過 `call by reference` 及 `return by reference` 達到連續運作的功能。例如: `int x,y,z; cin>>x>>y>>z;`。
- ⑪. `cin>>x;` 是 `cin.operator>>(x)` 或是 `operator>>(cin,x)` 的縮寫。
- ⑫. 宣告陣列 `int data[10][20];` 後，`data[x][y]` 與 `*(data+20*x+y)` 同義。
- ⑬. 在類別中，靜態成員函數可以使用成員變數。
- ⑭. 在全域中，函數名稱可以相同，但參數型別組態必須不同。
- ⑮. 類別中，`bool operator<(int);` 與 `bool operator<(int)const;` 是兩個不同的函數。
- ⑯. 類別中宣告變數或函數，在未指明的情況下都是私有的。
- ⑰. 類別中成員函數之所以可以使用成員變數及函數是因為有一個隱含的變數叫 `that`。
- ⑱. `inline` 函數是將函數本身在該行實作，所以編譯成執行檔時，可以擁有多個相對應同功能的程式碼本體。
- ⑲. 物件建構時，是先建構父類別、物件的成員變數，再執行建構子大括號內的指令。
- ⑳. 有兩個來源不同的函數匯入工作時，如果函數組態完全相等時，可以使用 `namespace` 將之分別。
- ㉑. 成員函數名稱及參數型別完全相同時，其回傳型別可以不相同。
- ㉒. 樣板函數，是先編譯成二進位函式庫，在連結時再進行連結生成可執行檔。
- ㉓. 類別的靜態變數，該類別只有一份，所有該類別物件共用之。
- ㉔. 類別的靜態變數，需要特別為該變數安排記憶體。
- ㉕. 函數的靜態變數，宣告時即已經安排好記憶體。

銘傳大學 99 學年度轉學生招生考試

資訊管理學系、資訊工程學系

第四節

程式設計試題

(第 2 頁共 2 頁) (限用答案本作答)

可使用計算機 不可使用計算機

函數填充題

一、請完成以下函數，主要輸入兩個正整數，輸出其最大公因數。

```
#include<②⑥>
using namespace ②⑦;
void main(){
    int m,n;cin>>m>>n;
    while(②⑧)swap(m,n);
    cout<<endl<<"GCD="<<n;
}
```

二、請完成以下函數，主要輸入三個點，輸出該三角形的重心。

```
#include<iostream>
using ②⑨ std;
class CPoint{
private:
    ③⑩ ostream& operator<<(③①,③②);
    double m_x,m_y;
public:
    ③③ X(){return m_x;}
    ③④ Y(){return m_y;}
    CPoint(double x,double y):③⑤{m_y=y;}
    ③⑥ operator+(const CPoint& pt)const{
        return CPoint(m_x+pt.m_x,m_y+pt.m_y);
    }
    ③⑦ operator/(int d)const{ return CPoint(m_x/d,m_y/d); }
};
istream& operator>>(istream& is, ③⑧ pt){ ③⑨ is>>pt.X()>>pt.Y();}
ostream& operator<<(ostream& os,const CPoint& pt){
    ④⑩ os<<"("<<pt.m_x<<","<<pt.m_y<<")";
}
void main(){
    CPoint A,B,C;cin>>A>>B>>C;
    cout<<endl<<"重心 G 為"<< (A+B+C)/3 <<" 。";
}
```