

Ch. 0 C++ Review 參考答案

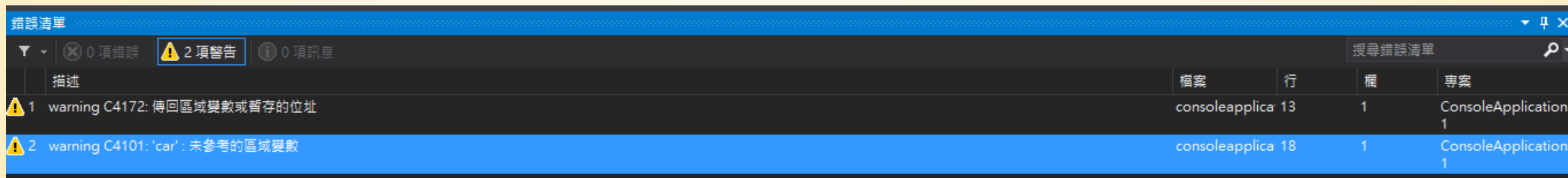
Question 1

- Please tell what's wrong with this following code

```
Car * foo() {  
    Car mazda ;  
    .....  
    return &mazda ;  
}
```

(Question 1) Ans:

- 一個是Car沒有被使用^[1]。
- 另外一個比較重要的是Car *foo()這個是function回傳的變數，其值及暫存位址，僅合法存在於Car *foo()這個function範圍內，出了這個function後，因為mazda變數無其他function承接，將會暫時存在於該function宣告此變數時的memory位置^[2]，等程式結束抑或是被其他function宣告變數時取代為止。
 - (補充) 如果您宣告區域範圍的物件，則會在傳回函式時終結該物件。如果函式傳回該物件的參考，則呼叫端嘗試使用 null 參考時，該參考可能會在執行階段造成存取違規。^[3]



錯誤清單				
0 項錯誤 2 項警告 0 項訊息				
描述		檔案	行	欄
1	warning C4172: 傳回區域變數或暫存的位址	consoleapplica	13	1
2	warning C4101: 'car' : 未參考的區域變數	consoleapplica	18	1

[1]C4101: <https://docs.microsoft.com/zh-tw/cpp/error-messages/compiler-warnings/compiler-warning-level-3-c4101?view=vs-2019>

[2]C4172: <https://docs.microsoft.com/zh-tw/cpp/error-messages/compiler-warnings/compiler-warning-level-1-c4172?view=vs-2019>

[3]傳回區域變數位址注意事項: <https://docs.microsoft.com/zh-tw/cpp/cpp/reference-type-function-returns?view=vs-2019>

Question 2

- Please show the output of the following code and explain what is going on.

```
#include<iostream>
using namespace std;

class Base{
    virtual void method(){
        cout<<"From Base"<<endl;
    }
public:
    virtual ~Base() {
        method();
    }
    void baseMethod(){
        method();
    }
};
```

```
class A : public Base{
    void method(){
        cout<<"from A"<<endl;
    }
public:
    ~A(){
        method();
    }
};

int main(){
    Base* base=new A();
    base->baseMethod();
    delete base;
    return 0;
}
```

(Question 2) Ans:

- 首先執行的是 `baseMethod()`，其會呼叫`method()`。雖然`base`宣告為`Base class`，但配置記憶體的类型為 `A class`，因此 `Base`中有加`virtual`的 `method`會被 `A`的 `method`取代，輸出 `from A`。
- 接著執行 `delete base`。由於 `A`是`Base`的`derived class`，因此在執行其`destructor`時，會先執行`derived class`（i.e., `A`）的`destructor`，之後再執行`base class`（i.e., `Base`）的`destructor`。因此執行結果為 `from A`再接著 `From Base`。
- **Output:**
 - `from A`
 - `from A`
 - `from Base`

Question 3

- Please explain the difference of the following two codes. In addition, can you give us examples of each code?

```
int (*Mul)(int a, int b);
```

```
int *Mul(int a, int b);
```

(Question 3) Ans: (1/2)

- `int (*mul) (int a,int b)` is function pointer

例子：將相乘函式 `mul()` 的位置傳入 `change()` 再進行乘法。

```
#include <iostream>
using namespace std;
int mul(int x,int y){
    return x*y;
}
int change(int a,int b,int (*func)(int,int)){
    return (*func)(a,b);
}
int main(){
    cout << change(3,4,&mul);
    return 0;
}
```

(Question 3) Ans: (2/2)

- `int *mul (int a,int b)` is just a simple declaration but return pointer.

例子：輸入行數和列數，輸出陣列開頭的指標，最後將陣列初始化為1。

```
#include <iostream>
using namespace std;
int *mul(int a,int b){
    int *p=new int[a*b];
    return p;
}
```

```
int main(){
    int row=4,col=3;
    int *p=mul(col,row);
    for(int i=0;i<col;i++){
        for(int j=0;j<row;j++){
            *(p+i*row+j)=1;
        }
    }
    return 0;
}
```


Question 4

- Explain what is “namespace pollution”.

(Question 4) Ans:

- Namespace pollution is a lot like pollution in general. It means that something is misplaced.
- 在library中，定義的global object與function必須是唯一，因此程式設計師會去避免使用到相同的名稱。當運用到libraries或source code files越來越多時，可能會造成程式設計師在選取名稱上的困擾，這個現象稱之為global name space pollution。

Question 5

- What is the output of the following code?

```
int i = 12;  
for(int i = 0; i<10; i++){  
    cout << i << endl;  
}  
cout << i << endl;
```

(Question 5) Ans:

0
1
2
3
4
5
6
7
8
9
12

(概念: Scopes <https://www.geeksforgeeks.org/scope-of-variables-in-c/>)

Question 6

- What is the output of the following code?

```
void func(int a, int& b, int* c){  
    a += 4;  
    b += 10;  
    *c = a;  
}  
int main(){  
    int a = 3, b = 5;  
    int *c = &b;  
  
    func(a, b, c);  
    cout << a << endl;  
    cout << b << endl;  
    cout << *c << endl;  
}
```

(Question 6) Ans:

Output:

3

7

7

(概念: Call by value or Call by reference

<https://www.guru99.com/call-by-value-vs-call-by-reference.html>)

Question 7

- What is the output of the following code?

```
int func(int a, int b){  
    return b+2;  
}  
int func(int* a, int &b){  
    return ++b;  
}  
int func(int a, int* b){  
    return a+2;  
}  
int main(){  
    int a=0, b=0;  
    cout << func(a,b) << endl;  
    cout << func(&a, b) << endl;  
    cout << func(a, &b) << endl;  
    cout << b << endl;  
}
```

Ans: Question 7

2

1

2

1

(概念: Function Overloading

<https://www.mygreatlearning.com/blog/function-overloading-in-c/>)

Question 8

- Write a piece of code to generate a 3x4 2-dimensional array using “new”.

(Question 8) Ans:

```
int **a=new int*[3];  
for(int i=0;i<3;i++){  
    a[i]=new int[4];  
}
```

(概念: Dynamic Memory Allocation

https://www.tutorialspoint.com/cplusplus/cpp_dynamic_memory.htm)

Question 9

- Write down your code to delete the 2-dimensional array created by using “delete”.

(Question 9) Ans:

```
int **array;  
array = new int* [size];  
  
for (int i = 0; i < size; i++)  
    array[i] = new int [K];  
  
for (int i = 0; i < size; i++)  
    for (int j = 0; j < K; j++)  
        array[i][j] = -1;  
  
for (int i = 0; i < size; i++){  
    delete [] array[i];  
}  
  
delete [] array;
```

参考:

<https://stackoverflow.com/questions/30720594/deleting-a-dynamically-allocated-2d-array>

Question 10

- Which line may possibly cause error? What is the reason? How to fix it?

```
class Student{
public:
    Student(int h, int
w):height(h),weight(w){};
    Student(const Student& _src){
        height = _src.height;
        weight = _src.weight;
    }
private:
    int height;
    int weight;
};
```

```
int main(){
    Student s(170, 60);
    Student w;
    Student t(s);
    return 0;
}
```

(Question 10) Ans:

- 宣告student w時會出現錯誤，因為沒有匹配的constructor可以呼叫，為解決這種狀況，可以替第一個 constructor 中的h 和w 加上初始值以解決這個問題。

```
Student(int h=0,int w=0): height(h),weight(w){}
```

Question 11

To make the output be 1 to 5, what should the numbers denoted by the red underlines be?

```
class A{
    public:
        A(){cout<<"__"<<endl;}
        ~A(){cout<<"__"<<endl;}
};
class B: public A{
    public:
        B(){cout<<"__"<<endl;}
        ~B(){cout<<"__"<<endl;}
};
class C:public B{
    public:
        C(){cout<<"__"<<endl;}
};
int main(){
    C c;
}
```

(Question 11) Ans:

```
class A{
    public:
        A(){cout<<1<<endl;}
        ~A(){cout<<5<<endl;}
};
class B: public A{
    public:
        B(){cout<<2<<endl;}
        ~B(){cout<<4<<endl;}
};
class C:public B{
    public:
        C(){cout<<3<<endl;}
};
int main(){
    C c;
}
```


Question 12

- Implement a class Complex, which represents the Complex Number data type. Implement the following operations
 - a) A constructor (including a default constructor which creates the complex number $0+0i$)
 - b) Overload operator+ to add two complex numbers
 - c) Overload operator* to multiply two complex numbers
 - d) Overload << and >> to print and read complex numbers. To do this, you will need to decide what you want your input and output format to look like.
 - e) Write a program according to the following specifications: use the constructor to define two complex numbers: $3+2i$ and $0+0i$. Input two complex numbers $5+3i$ and $0+0i$ using cin. Obtain the sum and product of all four complex numbers using operators + and *, respectively. Output the results using cout.

(Question 12) Ans: (1/2)

```
#include<iostream>
using namespace std;
class complex
{
public:
    float real,img;
    complex()
    {
        real=0;
        img=0;
    }
    complex operator +(complex);
    complex operator *(complex);
    friend ostream &operator<<(ostream&,complex&);
    friend istream &operator>>(istream&,complex&);
};
```

```
complex complex::operator +(complex obj)
{
    complex temp;
    temp.real=real+obj.real;
    temp.img=img+obj.img;
    return (temp);
}
complex complex::operator *(complex obj)
{
    complex temp;
    temp.real=(real*obj.real)-(img*obj.img);
    temp.img=(real*obj.img)+(img*obj.real);
    return (temp);
}
istream &operator>>(istream& is,complex& obj)
{
    is>>obj.real;
    is>>obj.img;
    return is;
}
```

(Question 12) Ans: (2/2)

```
ostream &operator<<(ostream& os,complex& obj)
{
    os<<obj.real;
    os<<"+"<<obj.img<<"i";
    return os;
}
```

```
int main()
{
    complex a,b,c,d;
    cout<<"\n Enter real and imaginary part of first complex number:";
    cin>>a;
    cout<<a;

    cout<<"\n Enter real and imaginary part of second complex number:";
    cin>>b;
    cout<<b;

    cout<<"\n Arithmetic operations are :";
    c=a+b;
    cout<<"\n Addition is:"<<c;
    d=a*b;
    cout<<"\n Multiplication is:"<<d<<"\n";

    return 0;
}
```