物件導向程式設計

Class Scope & Accessing Class Members

Joseph Chuang-Chieh Lin
Dept. CSIE, Tamkang University

Platform

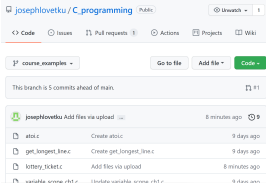
- Dev-C++

  Click here to download.

  Note: Please use this version otherwise you can't compile your programs/projects in Win10.

- OnlineGDB (https://www.onlinegdb.com/)

My GitHub page:
click the link here to visit.



- Other resources:
  - MIT OpenCourseWare - Introduction to C++ [link].
  - Learning C++ Programming [Programiz].
  - GeeksforGeeks [link]

Global Scope/Function Scope /Block Scope

Refer to: https://en.cppreference.com/w/cpp/language/scope
https://www.geeksforgeeks.org/scope-of-variables-in-c/

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

//global
int global = 5;

int main() {
    //local variable within a function scope
    int global = 2;
    cout << global << endl;

    return 0;
}
```

Global variables can be assessed from ANY part of the program.

- Usually declared outside all of the functions or blocks.

Global Scope/Function Scope /Block Scope

Refer to: https://en.cppreference.com/w/cpp/language/scope
https://www.geeksforgeeks.org/scope-of-variables-in-c/

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

int number = 1;

int main ()
{
  cout << "Global number: " << number << endl;
  int number = 2;
  cout << "Local number: " << number << endl;
  {
    int number = 3;
    cout << "Block number: " << number << endl;
  }
  return 0;
}
```

```
Global number: 1
Local number: 2
Block number: 3
```

Variables defined in a block “{ }” can be seen inside that block!

*Namespace Scope

- Purpose of using namespaces: [reference link]
  - Organize code into logical groups.
    - Prevent name collisions (namespace pollution).
  - Suitable for teamwork in case collisions happen.

- Usage:

  `using namespace::name;`

- Note：

  Don't use `using` in a header file.

  (potential unexpected name collisions)

```
#include <iostream>
using std:cin;

int main()
{
     int i;
     cin >> i;
     cout << i; // error!;
     std:cout << i; // OK!
     return 0;
}
```

*Namespace Examples

https://www.geeksforgeeks.org/namespace-in-c/

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

namespace first
{
    int val = 500;
}

int val = 100;

int main()
{
    int val = 200;

    // These variables can be accessed from outside the namespace using the scope operator ::
    cout << first::val << endl;

    return 0;
}
```

*Namespace Examples (can be skipped so far)

https://en.cppreference.com/w/cpp/language/namespace

```
namespace Q
{
    namespace V   // V is a member of Q, and is fully defined within Q
    { // namespace Q::V { // C++17 alternative to the lines above
        class C { void m(); }; // C is a member of V and is fully defined within V
                               // C::m is only declared
        void f(); // f is a member of V, but is only declared here
    }

    void V::f() // definition of V's member f outside of V
                // f's enclosing namespaces are still the global namespace, Q, and Q::V
    {
        extern void h(); // This declares ::Q::V::h
    }

    void V::C::m() // definition of V::C::m outside of the namespace (and the class body)
                   // enclosing namespaces are the global namespace, Q, and Q::V
    {}
}
```

Namespace in the same file

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

namespace TKU {
    class Student{
    public:
        string name; int age;
        double height; double weight;
        string department;
    };
    class Professor{
    public:
        string name; int age;
        double height; double weight;
        string department;
    };
}
```

```
int main() {
    TKU::Student s1;
    s1.age = 20;
    s1.name = "Betty";
    return 0;
}
```

Namespace in different files

- Suppose that we have
  - `main.cpp`
  - Files for TKU:
    - `tku.h`
    - `tku.cpp`
  - Files for UQ
    - `uq.h`
    - `uq.cpp`

Codes on OnlineGDB

Class assignment (1%)
Modify the code (line 19) to have the following output:

```
Betty is playing in UQ!
Betty is studying hard in UQ!
```

Class Scope

- Every class defines its own new scope.
- Outside the class scope, ordinary **data and function members** may be accessed only through an **object**, a **reference**, or a **pointer** using a **member access operator**

  . or **->**

- We access **type members** from the class using the **scope operator : :**
- In either case, the name that follows the operator must be a member of the associated class.

Class Scope (Example)

(click to the code in my GitHub page)

```
class rectangle {
public:
     typedef int unit;
     void area();
     void set(unit wd, unit ht);
private:
     unit width;
     unit height;
};
```

```
void rectangle::set(unit wd, unit ht)
{
     width = wd;
     height = ht;
}
```

```
void rectangle::area()
{
  cout << "The area: " << width * height << endl;
}
```

```
int main()
{
     rectangle obj, *obj2; //creating object of rectangle class
     rectangle::unit x, y;
     cin >> x;
     cin >> y;
     obj.set(x, y);
     obj2 = &obj;
     obj.area();
     obj2->area();
     return 0;
}
```

Class Scope (Example)

(click to the code in my GitHub page)

```
class rectangle {
public:
     typedef int unit;
     void area();
     void set(unit wd, unit ht);
private:
     unit width;
     unit height;
};
```

```
void rectangle::set(unit wd, unit ht)
{
     width = wd;
     height = ht;
}
```

define the member function outside a class

```
void rectangle::area()
{
  cout << "The area: " << width * height << endl;
}
```

```
int main()
{
     rectangle obj, *obj2; //creating object of rectangle class
     rectangle::unit x, y;
     cin >> x;
     cin >> y;
     obj.set(x, y);
     obj2 = &obj; // this usage of & is special in C++
     obj.area();
     obj2->area();
     return 0;
}
```

When a local variable has the same name as a global variable...

(reference link)

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

int x;  // Global x

int main()
{
    int x = 10; // Local x
    cout << "Value of global x is " << ::x;
    cout << "\nValue of local x is " << x;
    return 0;
}
```

```
Value of global x is 0
Value of local x is 10
```

To access a static variable in a class

```
using namespace std;

class Test
{
    static int x;
public:
    static int y;

    void func(int x)
    {
        cout << "Value of static x is " << Test::x;

        cout << "\nValue of local x is " << x;
    }
};
```

```
// static members must be explicitly
// defined
int Test::x = 1;
int Test::y = 2;

int main()
{
    Test obj;
    int x = 3 ;
    obj.func(x);

    cout << "\nTest::y = " << Test::y;

    return 0;
}
```

```
Value of static x is 1
Value of local x is 3
Test::y = 2;
```

Another Example: `Circle`

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

class Circle
{
    private :
        double radius; // data members
    public:
        void setRadius(double r);
        double getArea(); //member functions
};

void Circle::setRadius(double r)
{
    radius = r;
}

double Circle::getArea()
{
    return 3.14 * radius * radius;
}
```

```
int main()
{
    Circle c1; //an object of class circle
    c1.setRadius(2.5); //call member function
                       // to initialize radius
    cout << c1.getArea(); //display the area
    return 0;
}
```

Exercise 02 (3%) - BOOK Specification

Define a class **`BOOK`** with the following specification

- Private members:
  - **`BOOK_NO`**: (int)
  - **`TITLE`**: 20 characters (string)
  - **`PRICE`**: float (price per copy)
  - **`TOTAL_COST(N)`** : float (a function calculating the total cost for `N` copies; `N` is passed as argument)

- Public member functions:
  - **`INPUT()`** : Function to read `BOOK_NO`, `TITLE`, and `PRICE`.
  - **`PURCHASE()`** : Function to ask the user to input the number of copies to be purchased. It invokes `TOTAL_COST()` and prints the total cost to be paid by the user.

Sample Input & Output

C:\Users\josep_programs\PL\book_purchase.exe

```
In INPUT():
Enter Book Title: 淡江資工讚讚讚
Enter Book Number: 2022030501
Enter price per copy: 999
In PURCHASE():
Enter number of copies to purchase: 10
Total cost: 9990
--------------------------------
Process exited after 26.15 seconds with return value 0
請按任意鍵繼續 . . .
```

The main function:

```
int main()
{
    BOOK obj;
    obj.INPUT();
    obj.PURCHASE();
    return 0;
}
```

Another Example

Define a class **`student`** with the following specification

- Private members:
  - **`studentID`**: (int)
  - **`name`**: 20 characters (string)
  - **`eng`**, **`math`**, **`phy`**: float
  - **`total`**: float (the sum of eng, math, and phy)
  - **`grades()`**: a function to calculate eng + math + phy with float return type.
- Public member functions:
  - **`Takedata()`**: Function to accept values for admno, sname, eng, science and invoke ctotal() to calculate total.
  - **`Showdata()`**: Function to display all the data members on the screen.

Sample Input & Output

```
C:\Users\josep\_programs\PL\students.exe
In Takedata():
Enter studentID: 693410001
Enter student name: 林金毛
Enter grades in English, Math, and Physics: 100 99 92
In Showdata():
StudentID: 693410001
Student Name: 林金毛
English: 100
Math: 99
Physics: 92
Total: 291
--------------------------------
Process exited after 30.34 seconds with return value 0
請按任意鍵繼續 . . .
```

The main function:

```
int main ()
{
    student obj ;
    obj.Takedata();
    obj.Showdata();
    return 0;
}
```

Class Exercise 03 (2%)

Define a class **`student`** with the following specification

- Private members:
  - **`studentID`**: (int)
  - **`name`**: 20 characters (string)
  - **`eng`**, **`math`**, **`phy`**: float
  - **`total`**: float (the sum of eng, math, and phy)
  - **`avg_grade`**: average of eng + math + phy (float).
- Public member functions:
  - **`Takedata()`**: Function to accept values for admno, sname, eng, science and invoke ctotal() to calculate total.
  - **`Showdata()`**: Function to display all the data members on the screen.
  - **`PassOrFail()`**: Function to display "**`pass`**" if `avg_grade` >= 60 or "**`fail`**". otherwise.

Sample Input & Output

C:\Users\josep_Study_Programming Language\C++\hw_2.exe

```
In Takedata()
Enter studentID: 693410001
Enter student name: 張大勇
Enter grades in English, Math, and Physics: 100 90 80
In Showdata()
StudentID:693410001
Student Name:張大勇
English:100
Math:90
Physics:80
Total:90
Pass

--------------------------------
Process exited after 19.95 seconds with return value 0
請按任意鍵繼續 . . .
```

The main function:

```
int main ()
{
    student obj ;
    obj.Takedata();
    obj.Showdata();
    obj.PassOrFail();
    return 0;
}
```