Chapter 2+5 :

(Beginning with C++) & (Class and object)

Reference : E. Balaguruswamy Object Oriented Programming With C++; chapter-2,5.

Topics:

- **Topic 1: Basics of class and object**
- **Topic 2: Static data member & static function**
- **Topic 3: Friend function**

Questions:

Topic 1: Basics of class and object

1. What is class and object?how objects are created?Describe the relation between class and object.
2. Describe the memory allocation of object .
3. How does a class accomplish data hiding?
4. Describe the mechanism of accessing member and member function:
    a) Inside the main program
    b) Inside the member function of same class
    c) Inside the member function of another class
5. How to access private member function of a class?
6. What are the characteristics of the member function?
7. What are the various function that can have access the private and protected member of a class?
8. What is local class? explain with example.
9. Write a class to represent the the time that includes the member function to perform the following:
    - Take inputs for time in hours and minutes

- Multiply time by scalar value
- Add two times
- Display the time in hours : minutes

10. Write a class to represent the the time that includes the member function to perform the following:
    - Take two separate times input in hours and minutes
    - Add two times
    - Display the time in hours : minutes

Topic 2: Static data member & static function

11. What is static data member?Mention the properties of static data member.
12. What is static member function?Mention the properties of static member function.
13. When you declare a member of a class static- explain with an example .
    or, Write a c++ program that introduces the use of static data member and member function.

Topic 3: Friend function

14. What is friend function?When you declare a function as a friend function?Write down the characteristics of friend function.
15. What is forward declaration? Why do we use forward declaration? “Forward declaration is needed in case of friend function”- justify your answer in case of suitable program. Or, Show with suitable code segment how friend function can be defined?
16. Write down the advantages(merits) and disadvantages(demerits/drawbacks) of friend function.
17. Write a program that performs swapping private data of classes using the friend function.

Q: 1 :: What is class and object? how objects are created? Describe the relation between class and object.

Answer :

Class : Classes are user-defined data types that act as the prototype or template or blueprint for individual objects.

Objects : Objects are instances or variables of a class created with specifically defined data. Objects can correspond to real-world objects or an abstract entity.

Creating object : Once a class has been declared, we can create variables of that type by using the class name (like any other built-in type variable). So the class type variable is known as an object. For example,here “item” is a class where t1 and t2 are two objects of the class “ item”.

```
1   #include<iostream>
2   using namespace std;
3
4   class item
5   {
6       private:
7           int n;
8       public:
9           void getdata()
10          {
11              cin>>n;
12          }
13  };
14
15  int main()
16  {
17      item t1,t2;
18      t1.getdata();
19      return 0;
20  }
```

Objects can also be declared when a class is defined as follows:

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

class item
{
    private:
        int n;
    public:
        void getdata()
        {
            cin>>n;
        }
}t1,t2;

int main()
{
    t1.getdata();
    return 0;
}
```

Relation between class and object : Class and object are the building blocks of OOP. Space for member variables of the class are allocated separately for each object when they are created but before creating an object no memory space is allocated only for the class i.e. member variables. With objects we can access the public members of a class using the dot (.) operator. So without creating object of a class we can not use the class in the program.On the other hand we can not create any object without any class.So they are related to each other.

Q : 2::Describe the memory allocation of object .

Answer:

The memory space for objects is allocated when they are declared and not when the class is specified because class specification like a structure provides only a template and does not create any memory space for the objects. This statement is partly true. Actually, the member functions are created and placed in the memory space only once when they are defined as a part of a class specification. Since all the objects belonging to that class use the same member functions, no separate space is allocated for member functions when the objects are created. Only space for member variables is

allocated separately for each object. Separate memory locations for the objects are essential, because the member variables will hold different data values for different objects.

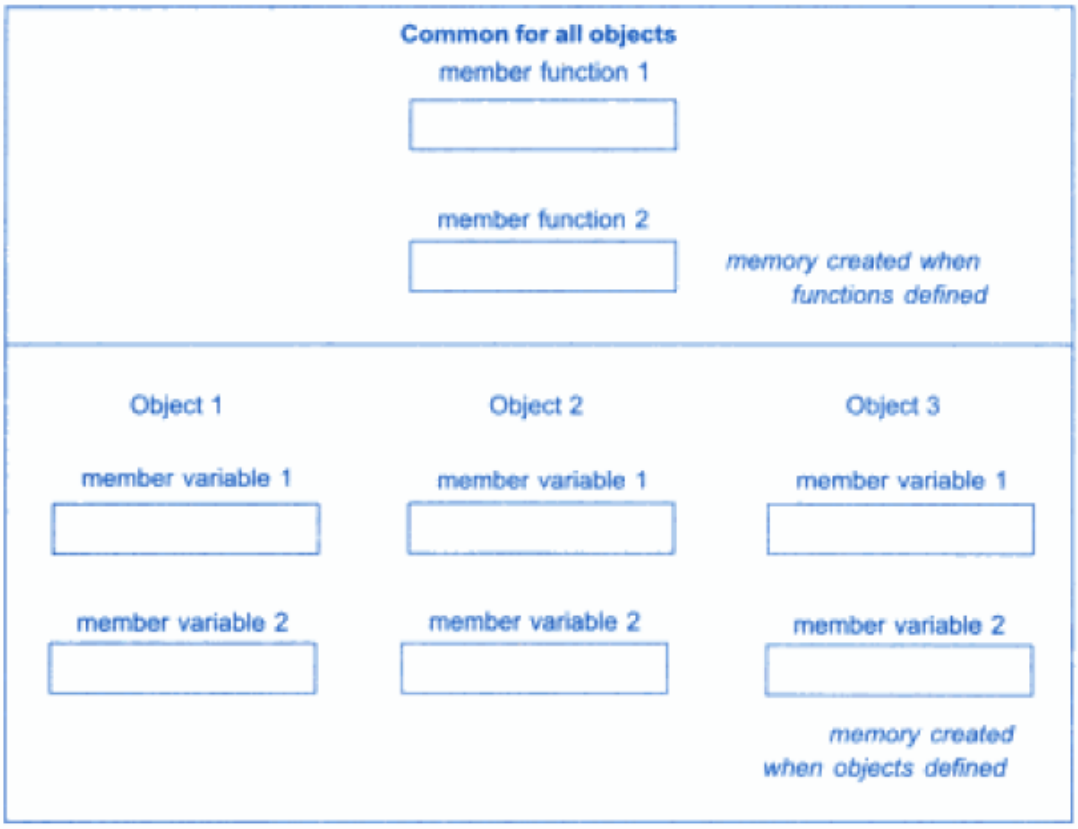


Fig : Objects memory allocation

Q : 3 :: How does a class accomplish data hiding(encapsulation/data binding)?

Answer :

The binding or wrapping up of data and functions, into a single unit (called class) is known as encapsulation. Data encapsulation is the most striking feature of a class,The data is not accessible to the outside world, and only those functions which are wrapped in the class can access it. These functions provide the interface between the object's data and the program.This insulation of the data from direct access by the program is called data hiding or information hiding.

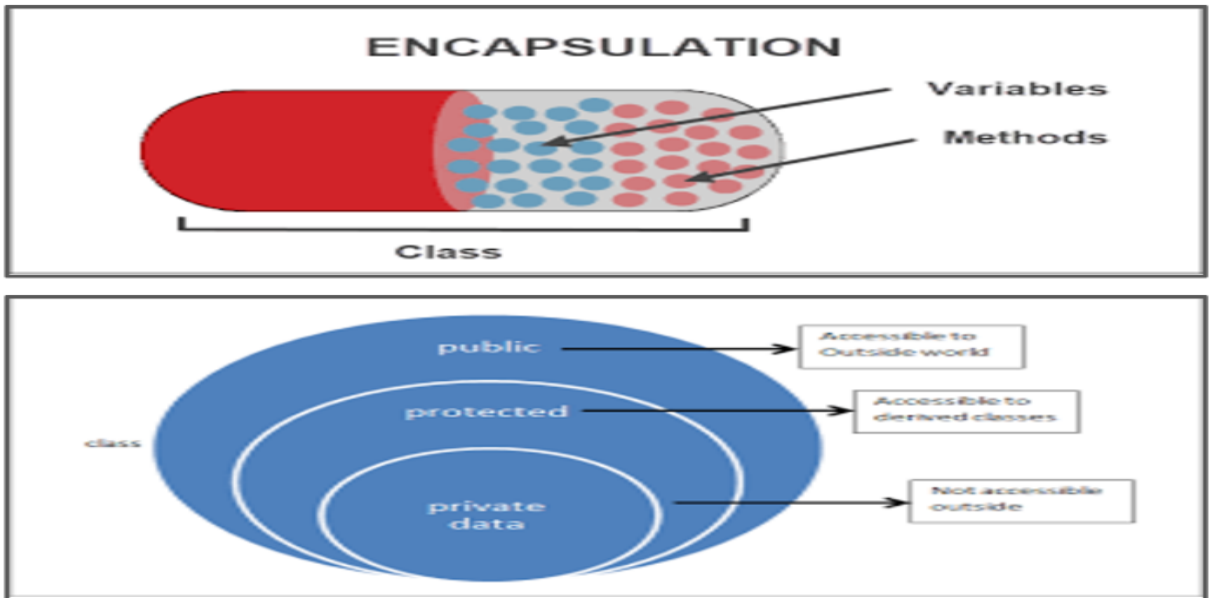


Fig : Encapsulation(data or information hiding/binding)

Within a class the members(data and function) are usually grouped under two sections, namely, private and public to denote which of the members are private and which of them are public. The keywords private and public are known as visibility labels. Note that these keywords are followed by a colon. Another keyword “protected” is used where this is utilized in the inheritance concept.The class members that have been declared as private can be accessed only from within the class. On the other hand, public members can be accessed from outside the class also.The data hiding (using private declaration) is the key feature of object-oriented programmIng. The use of the keyword private is optional. By default, the members of a class are private. If both the labels are missing, then, by default. all the members are private, Such a class is completely hidden from the outside world and does not serve any purpose. Any member function(private/public) can access the private members(data or function) but the private members(data or function) can not be accessed from outside the class. Only the public members(data or function) can be accessed from outside the class. In shorts,

- Encapsulating or binding data and functions within a class, to hide those from the rest of the program.
- Private section members(data and function)can be accessed by any member function of that class, not from the outside of that class.

- Protected section members(data and function)can be accessed by only the inherited classes member functions not from the outside of those classes.
- Public section members(data and function)can be accessed by the outside of that class.

Q :4:: Describe the mechanism of accessing member and member function:

a) Inside the main program

b) Inside the member function of same class

c) Inside the member function of another class

Answer:

a) Inside the main program : First of all we have to create an object of the class inside the main function(program) then by using dot operator(.) we can access the data member and member function of that class.

b) Inside the member function of the same class : Within a member function we can use data member and also can call the another member function of the same class as we regularly do in our normal function.

c) Inside the member function of the another class : Two way this type of access can be accomplished :

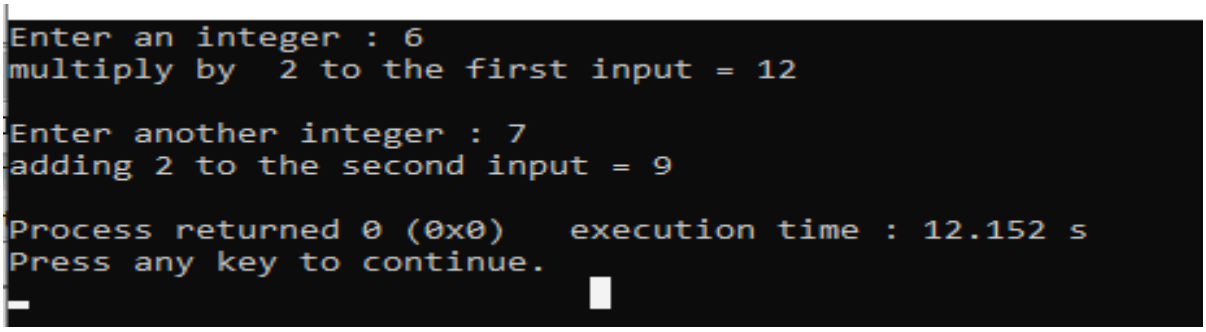
***i) In general process* :** Within a member function of a class(let class B) we can access the data member and member function of another class(let class A) by creating the object of this class(class A) with in the member function of the current class(class B), after that we can use the dot operator(.) with created object to access the data member and member function of that class(class A).Here it is important to mention that the accessing member related class(class A) should be properly define before creating object of that class within the current defined class(class B),otherwise it will generate error.

***ii) In inheritance based process* :** In inheritance based object oriented programming, within the member function of the sub-class (inherited class / derived class)by creating object of the base class(super class) we can access the data member and member function of that base class(super class).

Here is an example program(in c++) which will demonstrate all of our discussed things:

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


class A
{
    private:
        int n;
    public:
        void getdata();
        void putdata(int m)
        {
            cout<<"multiply by  2 to the first input = "<<m*2<<endl<<endl;
            getdata(); //b) Inside the member function of the class A access the member function of the same class
        }
};
class B
{
    private:
        int n;
    public:
        void getdata()
        {
            int m;
            cout<<"Enter an integer : ";
            cin>>n;
            m=n;
            A a1;
            a1.putdata(m); //c) Inside the member function of class A access member function of class B (another class)
        }
        void putdata(int m)
        {
            cout<<"adding 2 to the second input = "<<m+2<<endl;
        }
};
void A :: getdata()
{
    int m;
    cout<<"Enter another integer : ";
    cin>>n;
    m=n;
    B b1;
    b1.putdata(m); //c) Inside the member function of class A access member function of class B (another class)
}
int main()
{
    B t;
    t.getdata(); //a) Inside the main program access member function of class B
    return 0;
}
```

Output of the example program:



Github code - https://github.com/SyedaJannatul/DIIT_Object-Oriented-Programming/blob/b72b3495112fd189ffa7d763437568f1099f6a72/chapter%205_class%20and%20object/chapter%205_class/question4_code.cpp

Q :5:: How to access private member function of a class?

Answer :

In a class the private section members(data and function) can be accessed by only any member function of that class, not from the outside of that class, but the public section members(data and function) can be accessed by the outside of that class along with any member functions in the public section of that class. So to access the private members(data and function) of a class within that class we can use any member of that class.But if we want to access the private members(data and function) of a class from outside of that class then we have bring the private members(data and function) into the public section of that class,since only public section of a class is accessible from the outside of that class. Here is an example program:

```cpp
#include<iostream>
using namespace std;
class A
{
    private:
        int n;
        void getdata()
        {
            cout<<"Enter an integer : ";
            cin>>n; //access the private
                    //data member 'n' from private section
        }
    public:
        void putdata()
        {
            getdata();    //access private data member 'n' and
                         //member function'getdata()'from
                         //public section & bring public
            cout<<"entered integer + 5 = "<<n+5<<endl<<endl;
        }
};
int main()
{
    A t;
    t.putdata(); //access the private data member 'n' and
    return 0;    //member function 'getdata()' from outside the class
}
```

Output of the program :

```
Enter an integer : 6
entered integer + 5 = 11


Process returned 0 (0x0)   execution time : 3.079 s
Press any key to continue.
```

Q :6::What are the characteristics of the member function?

Answer:

- Member functions are used to generally leverage the data member of the class.
- Member function can be defined inside the class as regular function.
- Member function can be defined outside the class.To do so first we have to declare it within the class and then we can define it outside the class and before the main function as follows :

Function declaration with class:

```
Function_return_type        function_name(parameters);
```

Function defination outside the class:

```
Function_return_type        Class_name :: function_name(parameters)
{
        Statements;
}
```

- Several different classes can use the same functIon name. The membership label will resolve their scope.
- Member functions can access the private data of the class. A non- member function cannot do that.
- A member function can call another member function of the same class directly without using the dot operator.
- Private section member functions can be accessed by any member function of that class,but not from the outside of that class.
- Protected section member functions can be accessed by only the inherited classes member functions not from the outside of those classes.
- Public section members function can be accessed by the outside of that class along with any member functions in the public section of that class.

Q :7 :: What are the various function that can have access the private and protected member of a class?

Answer:

The function which can access the private members (data and function) of a class :

A. Public section member function of that class
B. Friend function which is declared within that class

The function which can access the protected members (data and function) of a class :

A. Private section or Protected section member functions of the derived class as the derived class mention the inheritance type from the base class.
B. Friend function which is declared within that class

C. Protected members that are also declared as static are accessible to any friend or member function of a derived class. Protected members that are not declared as static are accessible to friends and member functions in a derived class only through a pointer to, reference to, or object of the derived class.

Q :8 :: What is local class? explain with example.

Answer:

It is possible to define and use a class by creating an object of that class inside a function,such a class is called a local class.

Local classes can use global variables (declared above the function) and static variables declared inside the function but cannot use automatic local variables of the function. The global variables should be used with the scope operator (::). There are some restrictions in constructing local classes :

I. they cannot have static data members
II. member functions must be defined inside the local classes
III. the enclosing function cannot access the private members of a local class. However, we can achieve this by declaring the enclosing function as a friend.

Github code - https://github.com/SyedaJannatul/DIIT_Object-Oriented-Programming/blob/b72b3495112fd189ffa7d763437568f1099f6a72/chapter%205_class%20and%20object/chapter%205_class/question8_code.cpp

```cpp
#include<iostream>
using namespace std;
int m = 5;
void add()
{
    static int s = 2;
    class A
    {
        private:
            int n;
        public:
            void getdata()
            {
                cout<<"Enter an integer : ";
                cin>>n;
            }
            void putdata(int c)
            {
                int g = ::m;
                cout<<"Entered integer : "<<n<<endl;
                cout<<"Static data of the function : "<<c<<endl;
                cout<<"Global data : "<<g<<endl<<endl;
                cout<<"Sum = "<<n+c+g<<endl;
            }
    };
    A a;
    a.getdata();
    a.putdata(s);
}
int main()
{
        add();
        return 0;
}
```

```
Enter an integer : 10

Entered integer : 10
Static data of the function : 2
Global data : 5

Sum = 17

Process returned 0 (0x0)   execution time : 4.276 s
Press any key to continue.
```

Q:9 :: Write a class to represent the the time that includes the member function to perform the following:

a) Take inputs for time in hours and minutes

b) Multiply time by scalar value

c) Add two times

d) Display the time in hours : minutes

Answer :

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

class time
{
    private:
        int hours,minutes,scalar;
    public:
        void getdata(int h,int m,int s)
        {
            hours = h;
            minutes = m;
            scalar = s;
        }
        void sum(time);
        void putdata();
};
void time::sum(time t1)
{
    int temp_h;
    minutes = (t1.hours*60) + t1.minutes;
    minutes = minutes*t1.scalar;
    hours = minutes/60;
    minutes = minutes%60;
}

void time::putdata()
{
    cout<<"Hours : Minutes = "<<hours<<" : "<<minutes<<endl;
}

int main()
{
    time t;
    int hr,mint,sc=3;
    cout<<"Enter a time in hour and minute : ";
    cin>>hr>>mint;
    cout<<endl<<"Here the using scalar value is : "<<sc<<endl<<endl;
    t.getdata(hr,mint,sc);
    cout<<"Before operation the entered time, ";
    t.putdata();
    t.sum(t);
    cout<<"After operation the sum [entered time +(entered time * scalar value)] of two
times , ";
    t.putdata();
    return 0;
}

```

Output of the program :

```
Enter a time in hour and minute : 2 20

Here the using scalar value is : 3

Before operation the entered time, Hours : Minutes = 2 : 20
After operation the sum [entered time +(entered time * scalar value)] of two times , Hours : Minutes = 7 : 0

Process returned 0 (0x0)   execution time : 3.258 s
Press any key to continue.
```

Github code -
https://github.com/SyedaJannatul/DIIT_Object-Oriented-Programming/blob/b72b3495112fd189ffa7d763437568f1099f6a72/chapter%205_class%20and%20object/chapter%205_class/question9_code.cpp

Q:10 :: Write a class to represent the the time that includes the member function to perform the following: a) Take two separate times input in hours and minutes
b) Add two times
c) Display the time in hours : minutes

Answer :

Github code -
https://github.com/SyedaJannatul/DIIT_Object-Oriented-Programming/blob/b72b3495112fd189ffa7d763437568f1099f6a72/chapter%205_class%20and%20object/chapter%205_class/question10_code.cpp

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

class time
{
    private:
        int hours,minutes;
    public:
        void getdata(int h,int m)
        {
            hours = h;
            minutes = m;
        }
        void sum(time,time);
        void putdata();
};
void time::sum(time t11,time t22)
{
    int temp_h;
    minutes = t11.minutes + t22.minutes;
    hours = minutes/60;
    minutes = minutes%60;
    hours = hours + t11.hours + t22.hours;
}

void time::putdata()
{
    cout<<"Hours : Minutes = "<<hours<<" : "<<minutes<<endl;
}

int main()
{
    time t1,t2,t3;
    int hr1,mint1,hr2,mint2;
    cout<<"Enter a time in hour and minute : ";
    cin>>hr1>>mint1;
    cout<<"Enter another time in hour and minute : ";
    cin>>hr2>>mint2;

    t1.getdata(hr1,mint1);
    t2.getdata(hr2,mint2);

    cout<<"Before operation the entered time, "<<endl;
    t1.putdata();
    t2.putdata();
    t3.sum(t1,t2);
    cout<<"After operation the sum of two times , ";
    t3.putdata();
    return 0;
}

```

```
Enter a time in hour and minute : 2 45
Enter another time in hour and minute : 3 30
Before operation the entered time,
Hours : Minutes = 2 : 45
Hours : Minutes = 3 : 30
After operation the sum of two times , Hours : Minutes = 6 : 15
```

Q : 11:: What is static data member?Mention the properties of static data member.

Answer: A data member of a class can be declared as a static and is normally used to maintain values common to the entire class. A static member variable has certain special characteristics. These are :

- It is initialized to zero when the first object of its class is created. No other initialization is permitted.While defining a static variable, some initial value can also be assigned to the variable.
- Only one copy of that member is created for the entire class and is shared by all the objects of that class, no matter how many objects are created.
- It is visible only within the class, but its lifetime is the entire program.
- the type and scope of each static member variable must be defined outside the class definition. This is necessary because the static data members are stored separately rather than as a part of an object.

Q : 12:: What is static member function?Mention the properties of its.

Answer: Static member function is a member function that is declared as static, has the following properties:

- A static function can have access to only other static members (functions or variables) declared in the same class,
- A static member function can be called using the clase name (instead of its objects) as : class_name :: function_name();

Q:13 :: When you declare a member of a class static- explain with an example .Or, Write a c++ program that introduces the use of static data member and member function.

Answer : Static variables are normally used to maintain values common to the entire class. For example, a static data member can be used as a counter that records the occurrences of all the objects.

```cpp
#include <iostream>
#include<string.h>

using namespace std;

class Student
{
   private:
        int rollNo;
        string name;
        int marks;
   public:
        static int objectCount;

        static void showcount()
        {
          cout<<"count = "<<++objectCount<<endl;
          if (objectCount<=1)
            cout<<"Enter at least another student record "<<endl;
          cout<<endl;
        }

       void getdata()
       {
          cout << "Enter roll number: "<<endl;
          cin >> rollNo;
          cout << "Enter name: "<<endl;
          cin >> name;
          cout << "Enter marks: "<<endl;
          cin >> marks;
       }

       void putdata()
       {
          cout<<endl;
          cout<<"Roll Number = "<< rollNo <<endl;
          cout<<"Name = "<< name <<endl;
          cout<<"Marks = "<< marks <<endl;
          cout<<endl;
       }
};

int Student::objectCount;

int main()
{
   Student s1;
   s1.getdata();
   s1.putdata();
   Student::showcount();

   Student s2;
   s2.getdata();
   s2.putdata();
   Student::showcount();

   Student s3;
   s3.getdata();
   s3.putdata();
   Student::showcount();


   cout << "Total objects created = " << Student::objectCount << endl;
   return 0;
}

```

Output of the program :

```
Enter roll number:
1
Enter name:
sumi
Enter marks:
89

Roll Number = 1
Name = sumi
Marks = 89

count = 1
Enter at least another student record

Enter roll number:
25
Enter name:
rifat
Enter marks:
63

Roll Number = 25
Name = rifat
Marks = 63

count = 2

Enter roll number:
45
Enter name:
sohag
Enter marks:
78

Roll Number = 45
Name = sohag
Marks = 78

count = 3

Total objects created = 3

Process returned 0 (0x0)   execution time : 50.004 s
Press any key to continue.
```

Github code -

https://github.com/SyedaJannatul/DIIT_Object-Oriented-Programming/blob/b72b3495112fd189ffa7d763437568f1099f6a72/chapter%205_class%20and%20object/chapter%205_class/question13_code.cpp

Q: 14:: What is friend function?When you declare a function as a friend function?Write down the characteristics of friend function.

Answer:

Friend Function : Within a class the functions that are declared with the keyword friend are known as friend functions. A function can be declared as a friend in any number of classes. A friend function not a member function although it is declared as friend within class. It has full access rights to the private members of the class.

Necessity of friend function : We declare a function as a friend function when-

- more than one class shares a particular function,then to make the operation easier we make that particular function common for all the related classes by using the friend keyword with the function.
- need to access the private members(data and function) of the classes.
- Often used in operating overloading.

For example, consider a case where two classes, manager and scientist, have been defined. We would like to use a function income_tax() to operate on the objects of both these classes. In such situations, C++ allows the common function to be made friendly with both the classes, thereby allowing the function to have access to the private data of these classes. Such a function need not be a member of any of these classes.

Characteristics of friend function: A friend function possesses certain special characteristics:

- It is not in the scope of the class to which it has been declared as friend.
- Since it is not in the scope of the class, it cannot be called using the object of that class.
- It can be invoked like a normal function without the help of any object.
- Unlike member functions, it can not access the member names directly and has to use an object name and dot membership operator with each member name..
- It can be declared either in the public or the private part of a class without affecting its meaning.
- Usually, it has the objects as arguments.

Q :15 :: What is forward declaration? Why do we use forward declaration? "Forward declaration is needed in case of friend function"- justify your answer in case of suitable program. Or, Show with suitable code segment how friend function can be defined?

Answer :

Forward declaration : Forward declaration in C++ allows programmer to declare an identifier before it is defined. This means programmer can inform the compiler about a class, function, or variable before it is used in code. A declaration statement without a definition is used to achieve this.

Use of forward declaration : Forward declarations are important because they inform the compiler or interpreter what the identifier means, and how the identified thing should be used. A forward declaration may be optional or required, depending on the requirement. In case of friend function in c++ programming, forward declaration is required. The class names which will be used as parameters in the friend function declaration within the classes,need forward declaration,otherwise compiler will generate error.

When forward declaration of classes not used then in friend function declaration section, compiler will not understand the parameters type of the friend function as the parameters types are the classes name.

"Forward declaration is needed in case of friend function"- justify your answer in case of suitable program :

First write the use of forward declaration then write the following program

suitable code segment to demonstrate how friend function can be defined:

Github Code -

https://github.com/SyedaJannatul/DIIT_Object-Oriented-Programming/blob/b72b3495112fd189ffa7d763437568f1099f6a72/chapter%205_class%20and%20object/chapter%205_class/question15_code.cpp

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

class B;//class declaration or forward declaration

class A
{
    private:
        int data;
    public:
        void getdata(int n)
        {
            data=n;
        }
        friend void add(A,B);//friend function declaration
};

class B
{
   private:
        int data;
    public:
        void getdata(int n)
        {
            data=n;
        }
        friend void add(A,B);//friend function declaration

};

void add(A obj1 , B obj2)//friend function definition
{
    cout<<"Summation = "<<obj1.data+obj2.data;
}

int main()
{

    A a;
    B b;
    int p,q;
    cout<<"Enter two numbers : ";
    cin>>p>>q;
    a.getdata(p);
    b.getdata(q);
    add(a,b);
    return 0;
}

```

```
Enter two numbers : 4 5
Summation = 9
Process returned 0 (0x0)   execution time : 4.156 s
Press any key to continue.
```

Q:16:: Write down the advantages(merits) and the disadvantages (demerits / drawbacks) of friend function.

Answer:

Advantages:

1. Like any other function, the friend can be declared anywhere in the code.
2. It, like any other function, can be called without the usage of an object.
3. Many classes can benefit from the use of a function.
4. A friend function is used to access a class's non-public members.
5. Friend function enables the creation of more efficient programmes.
6. It adds extra functionality to the class that isn't commonly used.
7. It allows a non-member function to share private class information with another function.
8. It's utilized when two or more classes somehow have members tied to other portions of the program.

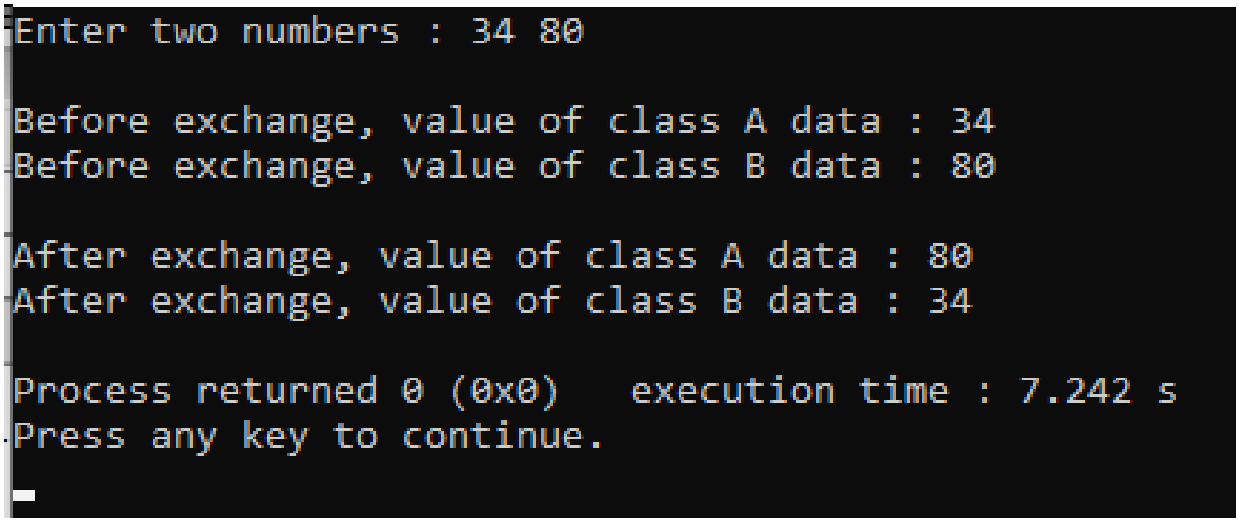
Disadvantages:

1. A derived class does not inherit a friend function.
2. Friend functions can not have a storage class specifier i.e they can not be declared as static or extern.
3. Allows private and protected members to be shown as the information of the class.
4. When a function is friend of more than one class then forward declaration of class is needed.

Q:17:: Write a program that performs swapping private data of classes using the friend function.

Answer :

```cpp
#include<iostream>

using namespace std;

class B;//class declaration or forward declaration

class A
{
    private:
        int data_A;
    public:
        void getdata(int n)
        {
            data_A=n;
        }

        void display()
        {
            cout<<data_A<<endl;
        }

        friend void swapping(A &,B &);//friend function declaration
};

class B
{
   private:
        int data_B;
    public:
        void getdata(int n)
        {
            data_B=n;
        }

        void display()
        {
            cout<<data_B<<endl;
        }

        friend void swapping(A &,B &);//friend function declaration

};

void swapping(A &obj1 , B &obj2)//friend function definition
{
    int temp;
    temp = obj1.data_A;
    obj1.data_A = obj2.data_B;
    obj2.data_B = temp;
}

int main()
{

    A a;
    B b;
    int p,q;
    cout<<"Enter two numbers : ";
    cin>>p>>q;
    a.getdata(p);
    b.getdata(q);

    cout<<endl;
    cout<<"Before exchange, value of class A data : ";
    a.display();
    cout<<"Before exchange, value of class B data : ";
    b.display();

    swapping(a,b);

    cout<<endl;
    cout<<"After exchange, value of class A data : ";
    a.display();
    cout<<"After exchange, value of class B data : ";
    b.display();

    return 0;
}
```



Github code -

https://github.com/SyedaJannatul/DIIT_Object-Oriented-Programming/blob/b72b3495112fd189ffa7d763437568f1099f6a72/chapter%205_class%20and%20object/chapter%205_class/question17_code.cpp