The friend function

Md. Sharif Hossen
Lecturer, Dept. of ICT, Comilla University, Bangladesh

Objectives

In this chapter, you will:

- How to use the friend function

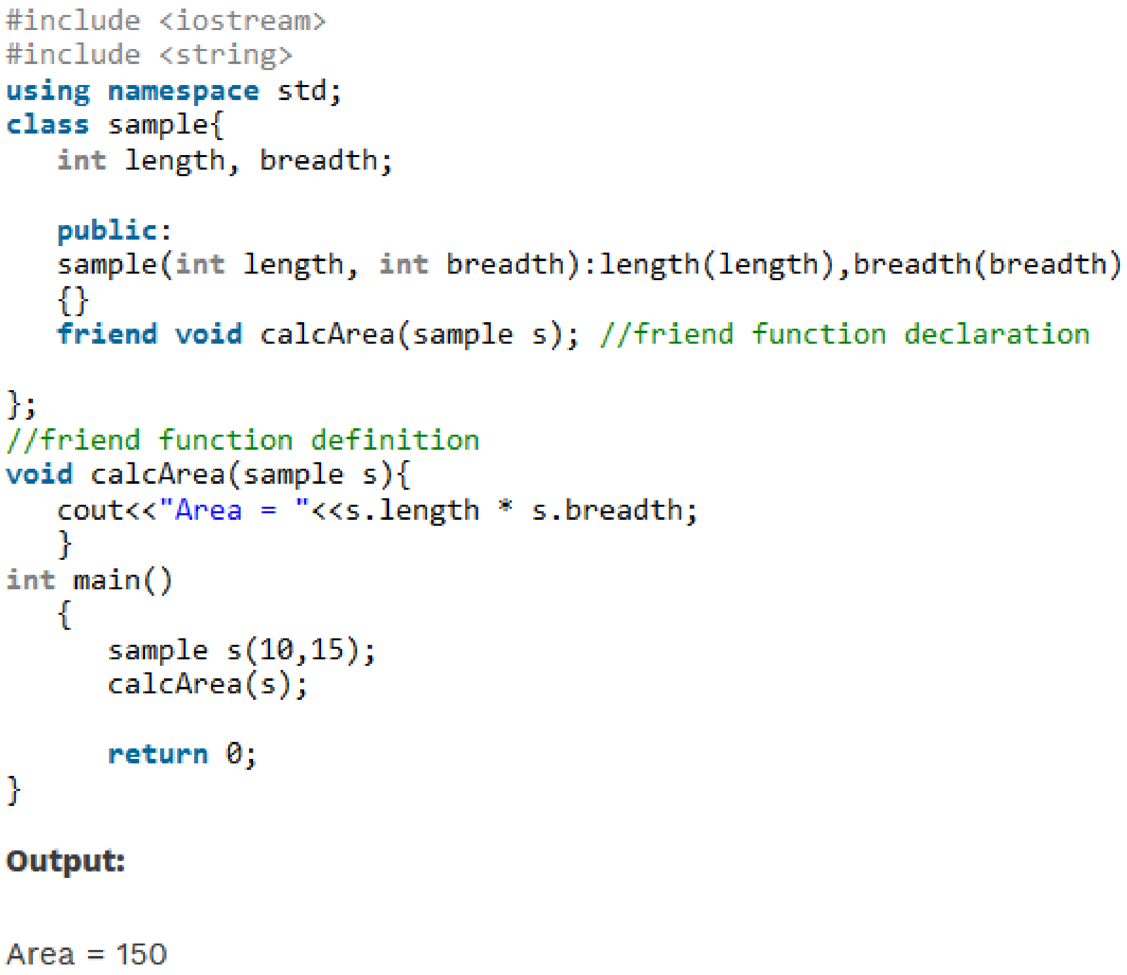
The friend function

A friend function in C++ is a function that is preceded by the keyword “friend”. When the function is declared as a friend, then it can access the private and protected data members of the class.

A friend function is declared inside the class with a friend keyword preceding as shown below.

```
class className{
    friend returnType functionName(arg list);
};
```

The friend function



```
#include <iostream>
#include <string>
using namespace std;
class sample{
   int length, breadth;

   public:
   sample(int length, int breadth):length(length),breadth(breadth)
   {}
   friend void calcArea(sample s); //friend function declaration

};
//friend function definition
void calcArea(sample s){
   cout<<"Area = "<<s.length * s.breadth;
   }
int main()
   {
      sample s(10,15);
      calcArea(s);

      return 0;
}
```

Output:

```
Area = 150
```

The friend class

Just like friend functions, we can also have a friend class. Friend class can access private and protected members of the class to which it is a friend.

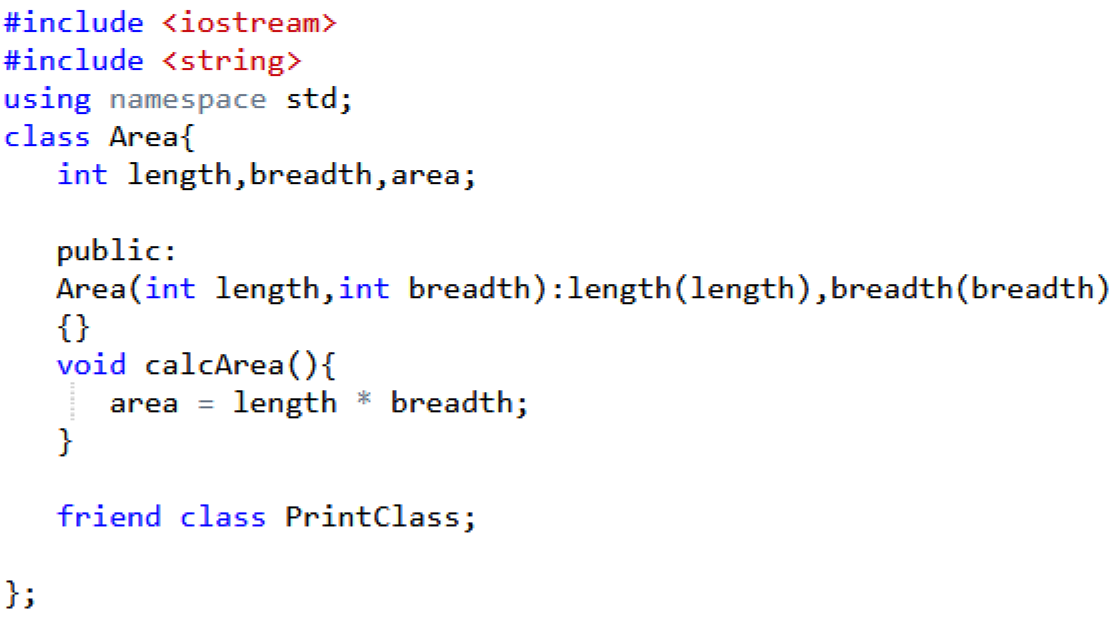
```
class A{
            ……
            friend class B;
};

class B{
            ……..
};
```

The class B is a friend of class A. So class B can access the private and protected members of class A.

But this does not mean that class A can access private and protected members of the class B.

The friend class

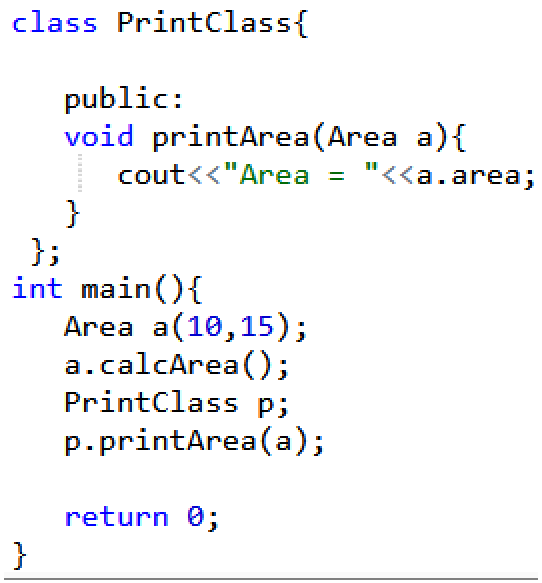


```
#include <iostream>
#include <string>
using namespace std;
class Area{
   int length,breadth,area;

   public:
   Area(int length,int breadth):length(length),breadth(breadth)
   {}
   void calcArea(){
      area = length * breadth;
   }

   friend class PrintClass;

};
```



```
class PrintClass{

   public:
   void printArea(Area a){
      cout<<"Area = "<<a.area;
   }
 };
int main(){
   Area a(10,15);
   a.calcArea();
   PrintClass p;
   p.printArea(a);

   return 0;
}
```