Friend function

- A ***friend function*** can access ***private and protected members of object,*** in which it declared as ***friend***
- A friend function can be:
  - ***A method*** of another class
  - A global function(***non-member function***)
- Following are some important points about friend functions.
  - ***Friends should be used only for limited purpose.***
  - Friendship is not mutual. If a class A is friend of B, then B doesn't become friend of A automatically.
  - Friendship is not inherited

e.g.

```
#include<iostream>
using namespace std;
class Demo
{
    private:
        int a ;
    protected:
        int b;
    public:
        int c;
};

main()
{
    Demo d;
    d.c = 10 ;
    cout << endl << "d.c is " << d.c;
    //d.b = 20 ;
    //d.a = 10 ;
}
```

e.g.

```
#include<iostream>
using namespace std;
class Demo
{
        private:
                int a ;
        protected:
                int b;
        public:
                int c;
                friend int main();
};

int main()
{
```

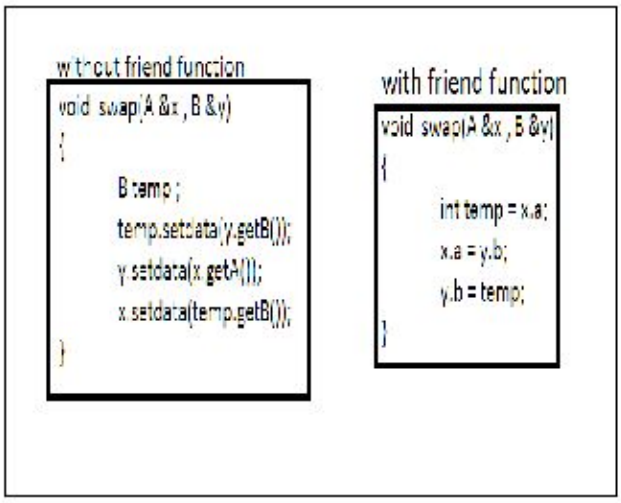
```
        Demo d;
        d.c = 10 ;
        cout << endl << "d.c is " << d.c;
        d.b = 20 ;
        d.a = 10 ;
        cout << endl << "d.a is " << d.a;
        cout << endl << "d.b is " << d.b;
}
```

- Friend function often used for ***coding simplicity.***

e.g.

```
#include<iostream>
using namespace std;
class B; //forward declaration
class A
{
        private :
                int a ;
        public:
                void setdata(int a)
                {
                        this->a = a ;
                }
                void display()
                {
                        cout << "\na = " << a ;
                }
                int getA()//inspector
                {
                        return a ;
                }
                friend void  swap(A &x , B &y);
};
class B
{
        private :
                int b ;
        public:
                void setdata(int b)
                {
                        this->b = b ;
                }
                void display()
                {
                        cout << "\nb = " << b ;
                }
                int getB()//inspector
                {
                        return b ;
                }
```

```
                friend void  swap(A &x , B &y);
};

void  swap(A &x , B &y)
{
        int temp = x.a;
        x.a = y.b;
        y.b = temp;
}

main ()
{
        A i ;
        B o ;
        i.setdata(10);
        o.setdata(20);
        cout <<endl <<" before swapping ";
        cout << endl << "object named i";
        i.display();
        cout << endl << "object named o";
        o.display();

        swap(i,o);

        cout <<endl <<" after swapping ";
        cout << endl << "object named i";
        i.display();
        cout << endl << "object named o";
        o.display();

}
```



- As we know friend function can be a Non - Member Function or ***It can be member function of other class.***
- If we want a friend function then ***its prototype must be place inside the class with friend keyword as prefix.***
- Once a function become a friend of a class, ***then function have special rights that it can access all the members whether they are public, protected or private by using . Operator of object.***

Friend class

- ***A class can also be a friend of another class.***
- If a class is a friend of another class that means ***all the members functions directly access private and protected member of another class.***

```
class A
{
```

```
        private :
                int a;
        public :
                friend class B;
};

class B
{
        public:
                void ABC()
                {
                        A x ;
                        int temp = x.a ;  // OK

                }
                void xyz()
                {
                        A x ;
                        int temp = x.a ;
                }
}
```