---

c++ :- not pure oops

c++ :- partial oops

NORMAL FUN. MEMBER FUNCTION (METHOD)

---------- ----------

fact() p . get();

OBJECT X OBJECT

(.) dot operator

---

friend function -> normal function -> access private data member

drawback of oops

```
      ADDTION OF TWO COMPLEX NOS USING FRIEND FUNCTION
      ------------------------------------------------

            RETURNING  OBJECT
            -----------------
*/

class  complex
{
      private : int   a , b;

      public :
            void   get()
            {
                  cout<<" ENTER REAL AND IMAG VALUE " << endl;
                  cin >> a >> b;
            }

            void out()
            {
                  cout<< a << "+i " <<  b << endl;
            }

friend complex sum ( complex x , complex y ); // DECLARE FRIEND FUN.
 };
```

```
complex      sum ( complex x , complex  y )  // normal fun.
 {
              complex  w ; // temparary object

              w . a = x . a + y . a ;    // w . a = 3 + 4 = 7

              w . b = x . b + y . b;   //   w . b = 2 + 3 = 5

              return( w );   // returning object
}

        int     main()
        {
                    complex p , q , t ; // OBJECT

                    p . get();    // 3 , 2

                    q . get();    // 4 , 3

                    t = sum ( p, q ); // NORMAL FUNCTION

                    p . out();  // 3 + i 2
                    q . out();  //  4 + i 3
                    t . out();  //  7 + i 5

         }
```

```
//FRIEND FUNCTION -> NORMAL FUNCTION -> ACCESS PRIVATE DATA MEMBER

#include<iostream>
using namespace std;

  class test2 ; // FORWARD DECLARATION

  class test1
  {
            private : int  a;

            public:
                        void get()
                        {
                              cout<< " ENTER a " << endl;
                              cin >> a;
                        }
                        void out()
                        {
                              cout << " a = " << a << endl;
                        }

friend  void swap (test1  &x , test2  &y);// declaration friend fun.

  };

  class   test2
```

```
{
        private : int  b;

        public  :
                    void get()
                    {
                        cout<< " ENTER b " << endl;
                        cin >> b;
                    }

                  void  out()
                  {
                        cout << " b = " << b << endl;
                  }

friend   void swap ( test1 &x , test2 &y); // declaration friend
fun.
 };

      void    swap ( test1 &x , test2  &y)
      {
            int c;

            c = x.a;    // c   = 5

            x.a = y.b;  //  x.a =  2

            y.b = c;   //   y.b = 5
```

```
}

int    main()
{
        test1 p;

        test2  q;

        p. get();   // p.a = 5

        q. get();   // q.b = 2

        swap( p,q ); // normal function

        p. out();   // p.a = 2

        q. out();   // q.b = 5
}
```

```
/********** Friend Class **********/

#include<iostream>
using namespace std;

class my
{
	private : int a,b;
	public:
	my(int x,int y)
	{
		a=x;
		b=y;
	}
	friend class you;
};

class you
{
	public:
	int max(my tmp)
	{
		return (tmp.a > tmp.b) ? tmp.a : tmp.b;
	}
};
int main()
{
	my p(4,7);
```

```
        you q;

        cout<<q.max(p);
}
```