friend Functions

friend Functions

- A class's **private** members are accessible to only its methods and its **friend** functions
- The **protected** members of a class are accessible to only the methods in its class hierarchy and its **friend** functions

Lets make **friends**

Use the keyword **friend**

```
class C  {
//....
friend int ff( );   //declare a friend function
//....
};
```

Note: ff( ) is not a method so it only serves to give ff( ) access rights to the class C's private and protected members

Too Many **friends**

- Yes, too many **friend** functions can be dangerous
  - friend functions go against the principles of OOP
  - They may also be placed within the private, protected, or public part of the class declaration

However, it's OK to make friends

- If you use friend functions, a recommendation would be to only use them in operator overloading

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

class OrdPair{
public:
  OrdPair( ){   //default
    p1 = 0.0;
    p2 = 0.0;
  }
  OrdPair( float f1 , float f2 ) {  //constructor
    p1 = f1;
    p2 = f2;
  }
  bool operator==(const OrdPair& ) const;  //prototypes

friend OrdPair operator*( OrdPair&, OrdPair& );  //declare the friend

  OrdPair operator-(const OrdPair& ) const;
  void write_it( ) const;
private:
  float p1, p2;   };

bool OrdPair::operator==(const OrdPair& s) const {
  return p1 == s.p1 && p2 == s.p2;  }

OrdPair operator*( OrdPair& s, OrdPair& t ) {  //The friend defined

  return OrdPair(s.p1*t.p1, s.p2*t.p2);  }

OrdPair OrdPair::operator-( const OrdPair& s) const {
  OrdPair z(p1 - s.p1, p2 - s.p2);
  return z;  }

void OrdPair::write_it( ) const {
  cout << "The result is: (" << p1 << "," << p2 << ")" << endl;
}

int main()
{
  OrdPair s1(32, 2), s2(32,-2), s3;
  if(s1.operator==(s2))
   // or can use
    //if (s1 == s2)
    cout << "equal" << endl;
  else
    cout<< "Not equal" << endl;
  //s3 = s1.operator*(s2);  // s3 = s1.operator-(s2);   //s3 = s1 - s2;
 s3 = s1 * s2;
 s3.write_it( );
  return 0;      }
```

friend Exercise

- Create a friend function for the addition, +, and subtraction, -, operations, whereby the x coordinate is added (respectively subtracted) with the x coordinate and the y with the y.