

```
#include <iostream>

using namespace std;

class A {
//private:
    int a;

public:
    A() { a = 0; }
    friend class B; // Friend Class
};

class B {
//private:
    int b;

public:
    void showA(A x)
    {
        // Since B is friend of A, it can access
        // private members of A
        cout<<"Enter value of x\n";
        cin>>x.a;
        cout << "A::a=" << x.a;
    }
};

int main()
{
    A a;
    B b;
    b.showA(a);
    return 0;
}
```

```
}
```

```
#include <iostream>
```

```
#include<string>
```

```
using namespace std;
```

```
class XYZ {
```

```
private:
```

```
    int num;
```

```
    char ch;
```

```
public:
```

```
    friend void disp(XYZ obj)
```

```
    {
```

```
        cout<<obj.num<<endl;
```

```
        cout<<obj.ch<<endl;
```

```
    }
```

```
    void read();
```

```
};
```

```
void XYZ::read()
```

```
{
```

```
    cout<<"Enter number";
```

```
    cin>>num;
```

```
    cout<<"Enter char";
```

```
    cin>>ch;
```

```
}
```

```
int main() {
```

```
    XYZ obj1;
```

```
    obj1.read();
```

```
    disp(obj1);
```

```
    return 0;
```

```
}
```

```

#include <iostream>

using namespace std;

class XYZ {
private:
    char ch;

    int num;
public:
    /* This statement would make class ABC
    * a friend class of XYZ, this means that
    * ABC can access the private and protected
    * members of XYZ class.
    */

    friend class ABC;
};

class ABC {
public:

    void data(XYZ obj){
        cout<<"Enter char";

        cin>>obj.ch;

        cout<<"Enter number";

        cin>>obj.num;

        cout<<"Char";

        cout<<obj.ch<<endl;

        cout<<"Number";

        cout<<obj.num<<endl;

    }

};

int main() {

```

```
    ABC obj;  
    XYZ obj2;  
    obj.data(obj2);  
    return 0;  
}
```

```
#include<iostream>  
using namespace std;  
    class Rectangle  
    {  
        int L,B;  
  
        public:  
        Rectangle()  
        {  
            L=10;  
            B=20;  
        }  
  
        friend class Square;    //Statement 1  
};  
  
class Square  
{  
    int S;  
  
    public:  
    Square()  
    {  
        S=5;  
    }  
}
```

```

        void Display(Rectangle Rect)
        {
            cout<<"\n\n\tLength : "<<Rect.L;
            cout<<"\n\n\tBreadth : "<<Rect.B;
            cout<<"\n\n\tSide : "<<S;
        }
    };

    int main()
    {

        Rectangle R;
        Square S;

        S.Display(R);    //Statement 2

    return 0;
    }

/* C++ program to demonstrate the working of friend function.*/
#include <iostream>
using namespace std;

class Distance
{
    private:
        int meter;
    public:
        Distance()
        {

```

```

        meter=0;
    }

    //friend function
    friend int addFive(Distance d);
};

// friend function definition
int addFive(Distance d)
{
    //accessing private data from non-member function
    d.meter += 5;
    return d.meter;
}

int main()
{
    Distance D;
    cout<<"Distance: "<< addFive(D);
    return 0;
}

#include <iostream>
using namespace std;

// forward declaration
class B;
class A {
    private:
        int numA;
    public:
        A()

```

```

        {
            numA=12;
        }

// friend function declaration
friend int add(A, B);
};

class B {
private:
    int numB;
public:
    B()
    {
        numB=1;
    }

// friend function declaration
friend int add(A , B);
};

// Function add() is the friend function of classes A and B
// that accesses the member variables numA and numB
int add(A objectA, B objectB)
{
    return (objectA.numA + objectB.numB);
}

int main()
{
    A objectA;
    B objectB;

    cout<<"Sum: "<< add(objectA, objectB);
}

```

```
return 0;
```

```
}
```