

Friends Function.

Properties of friend function.

- Not in the Scope of class
- Some Complex Number can do anything with Private Part.
- Since it is not in the Scope of class so it is not called the object of class.
- Usually Contains the objects as Argument

Friend class:

A Friend Class Can Access Private member of other Class. in which it is declare as Friend.

Example:-

```
class node {  
    int Key;  
    Node * next;  
    friend class LinkedList;  
}
```

Friend Function:

A Friend function is also used to access the Private

members of class.

Type:

- 1) A member of other class
- 2) A global function.

Example:

```
Class Node {
```

```
private:
```

```
int key;
```

```
Node * next;
```

```
friend int LinkedList::search();
```

Important points:

- 1) Friends should be used only for limited purpose.
- 2) Friendship is not mutual. if A is a friend of B then it doesn't mean B is a friend of A automatically.
- 3) Friendship is not inherited.
- 4) The concept of friend is not there in java.

Example:

```
Class A {
```

```
int a;
```

```
public:
```

```
A() { a = 0; }
```

```
friend class B;
```

```
};
```

```
int main ()
```

```
{
```

```
    A a;
```

```
    B b;
```

```
    b.ShowA(a);
```

```
    return 0;
```

```
class B {
```

```
int b;
```

```
public:
```

```
void ShowA(A &a)
```

```
{ cout << a.a;
```