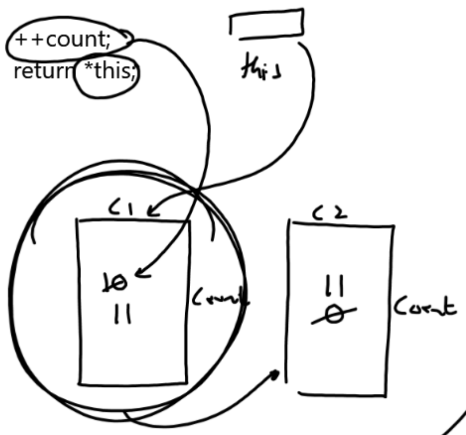
```
#include <iostream>
using namespace std;
class Counter
{
   int count;
public:
   Counter(int c)
   {
      count=c;
   }
   Counter()
   {
      count=0;
   }
   void show()
   {
      cout<<"Count:"<<count<<endl;
   }
   Counter operator++();
};
```

```
Counter Counter::operator++()
{
   ++count;
   return *this;
}
```



C2 = C1.op++();

```
int main()
{
✓  Counter C1(10);
✓  Counter C2;
   C1.show();     10
   C2.show();     0
   C2=++C1;
   C1.show();
   C2.show();
   return 0;
}
```

```
#include <iostream>
using namespace std;
class Counter
{
   int count;
public:
   Counter(int c)
   {
      count=c;
   }
   Counter()
   {
      count=0;
   }
   void show()
   {
      cout<<"Count:"<<count<<endl;
   }
   Counter operator++();
};
```

```
Counter Counter::operator++()
{
   ++count;
   return count;
}
```

11

How this code working? Explain!

```
int main()
{
   Counter C1(10);
   Counter C2;
   C1.show();
   C2.show();
   C2=++C1;
   C1.show();
   C2.show();
   return 0;
}
```

Overloading Of Post Increment Operator

```
#include <iostream>
using namespace std;
class Counter
{
   int count;
public:
   Counter(int c)
   {
      count=c;
   }
   Counter()
   {
      count=0;
   }
   void show()
   {
      cout<<"Count:"<<count<<endl;
   }
   Counter operator++(int);
};
```

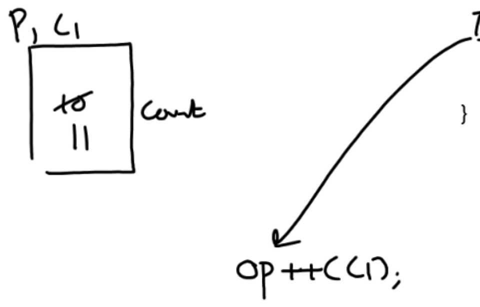
```
Counter Counter::operator++(int)
{
   Counter Temp;
   Temp.count=count;
   count++;
   return Temp;
}
```

```
int main()
{
   Counter C1(10);
   Counter C2;
   C1.show();
   C2.show();
   C2=C1++;
   C1.show();
   C2.show();
   return 0;
}
```

Overloading Pre Increment Operator Using Friend Function

```
#include <iostream>
using namespace std;
class Counter
{
   int count;
public:
   Counter(int c)
   {
      count=c;
   }
   Counter()
   {
      count=0;
   }
   void show()
   {
      cout<<"Count:"<<count<<endl;
   }
   friend void operator++(Counter &);
};
void operator++(Counter &P)
{
   ++P.count;
}
```

```
int main()
{
  ✓Counter C1(10);
   C1.show(); —> 10
   ++C1;
   C1.show(); —> 11
   return 0;
}
```



```
#include <iostream>
using namespace std;
class Counter
{
    int count;
public:
    Counter(int c)
    {
        count=c;
    }
    Counter()
    {
        count=0;
    }
    void show()
    {
        cout<<"Count:"<<count<<endl;
    }
    friend Counter operator++(Counter&);
};
Counter operator++(Counter &P)
{
    Counter Temp;
    ++P.count;
    Temp.count=P.count;
    return Temp;  }
```

```
int main()
{
    Counter C1(10);
    Counter C2;
    C1.show();
    C2.show();
    C2=++C1;
    C1.show();
    C2.show();
    return 0;
}
```