Wednesday, September 3, 2014 1:28 PM

Sequence.h

```
class sequence
{
    public:
        int x;
        void fun1()
        {
            fun2();
            y = 20;
        }
    private:
        int y;
        void fun2();
};
void sequence::fun2()
{
    x = y;
}
```

Sequence.cpp

```
#include "sequence.h"
```

main

```
sequence s;
s.x = 10;
s.fun1();
```

s.y = 20; error

s.fun2(); error

---

lecture 4

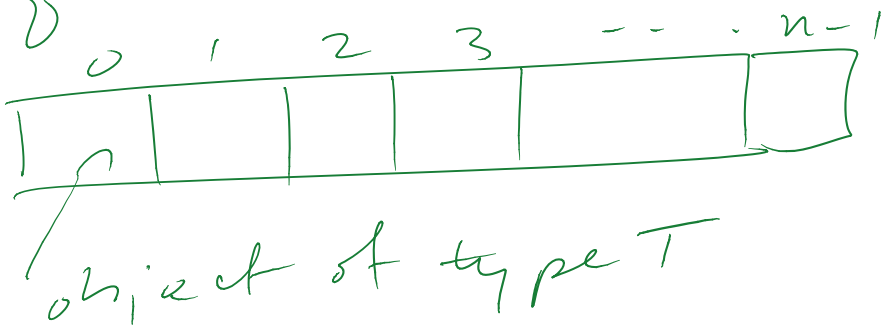
```
#include "sequence.h"
```

---

class structure

dynamic memory allocation

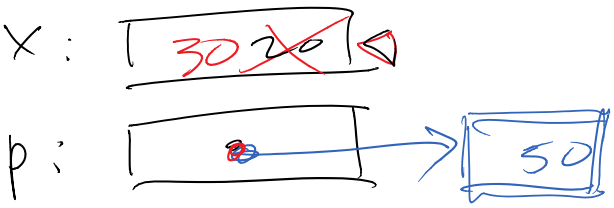
sequence



{

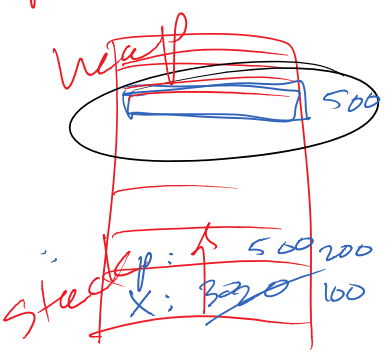
```
int x;
int* p;
p = &x;
x = 20;
*p = 30;
```

← automatic storage



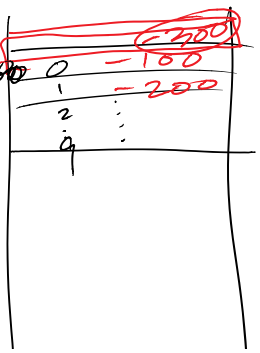
allocate space

```
p = new int;
*p = 50;
```



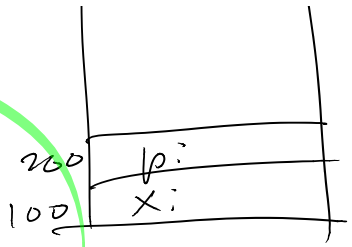
memory leak

```
p = int int[10];
*p = -100;
*(p+1) = -200;
```



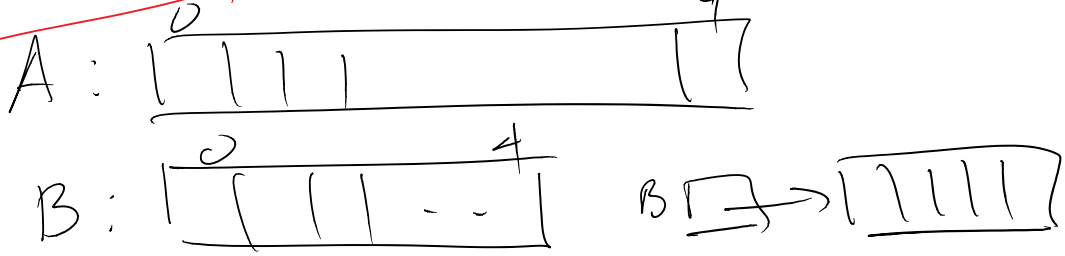
old school

```
*(p+1) = -200;
*(p-1) = -300;
```



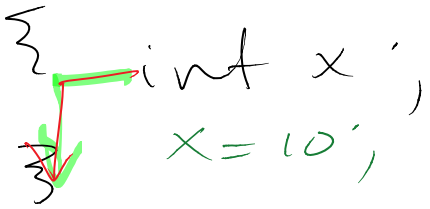

```
p[0] = -100;
p[1] = -200;
p[-1] = -300;
```

```
{ int A[10];
  int *B;
  B = new int[5];
```




```
  delete [] B;
}  — A released
```

```
{ int x;
  { static int x;
    { int x;
```


int x;
x=10;