CS 162
Intro to CS II

More Classes

Odds and Ends…

- Sign up for Assignment #1 demo
- Assignment #2 released
- Sign up for Exercise #3 study session
- Email late Assignment #1 to me

get two weeks, but do this sooner rather than later

Get started early!

"Poor planning on your part doesn't constitute an emergency on my end!!!"

I will acknowledge when I've uploaded and it's ready to demo

Class Type Member

```
class Point {
public:
  Point();   //Default Constructor
…
private:
  int x;
  int y;
};
class Points {
public:
  Point p;
};
int main() {
  Points pts;
  cout << pts.p.get_x();
  return 0;
}
```

"has a" or is "part of" relationship

notice two dots to access p's stuff.

- member of a class is an object

Points "has a" Point

or

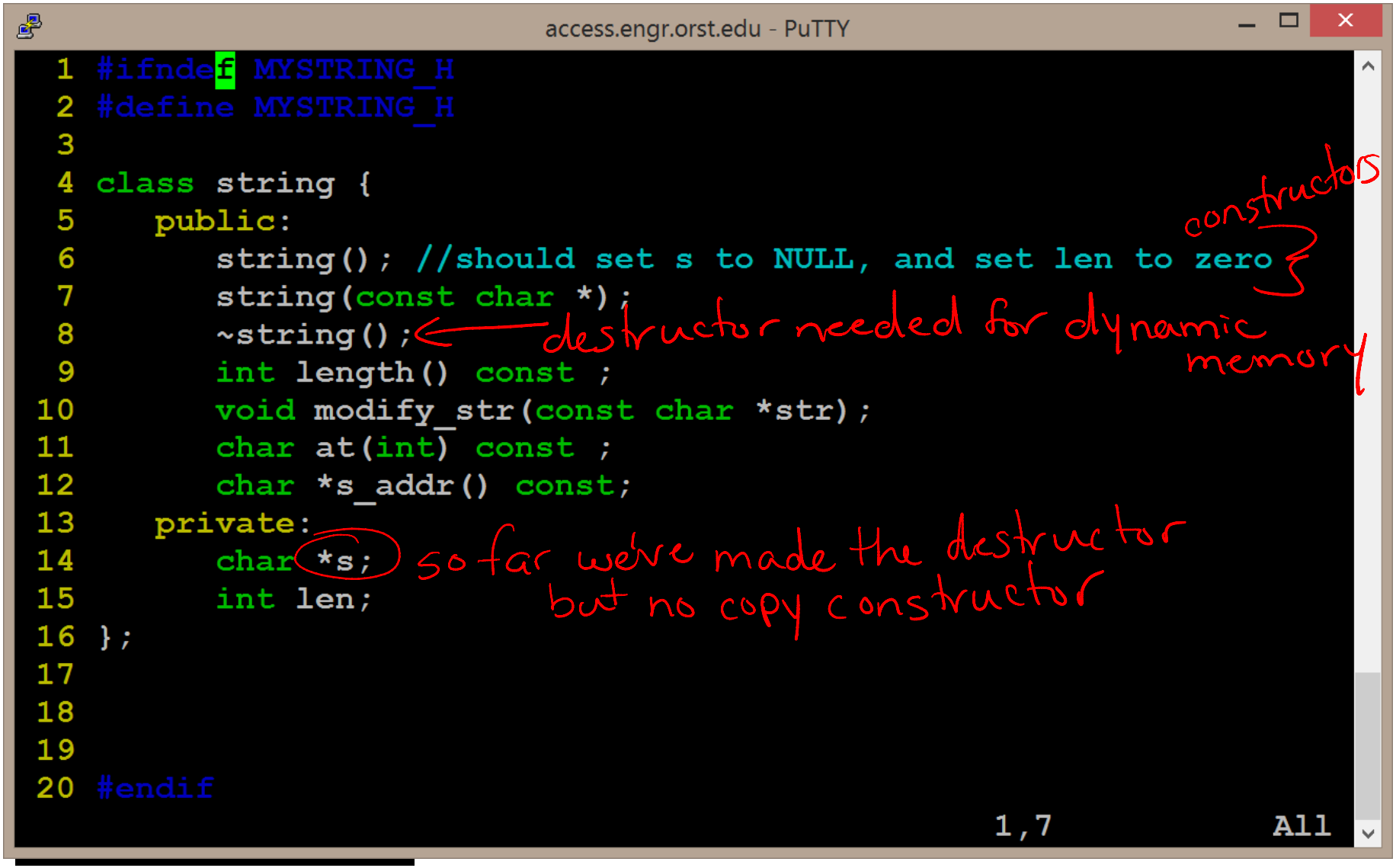
Point is "part of" Points

Class Type Member

```
class Point {
public:
   Point();   //Default Constructor
…
private:
   int x;
   int y;
};
class Points {
public:
   Points();   //Default Constructor
private:
   Point p;
};
Points::Points() : p() { }
```

need to call constructor for member object

Revisit Copy Constructors/Destructors



mystring.cpp has destructor but no copy constructor

access.engr.orst.edu - PuTTY

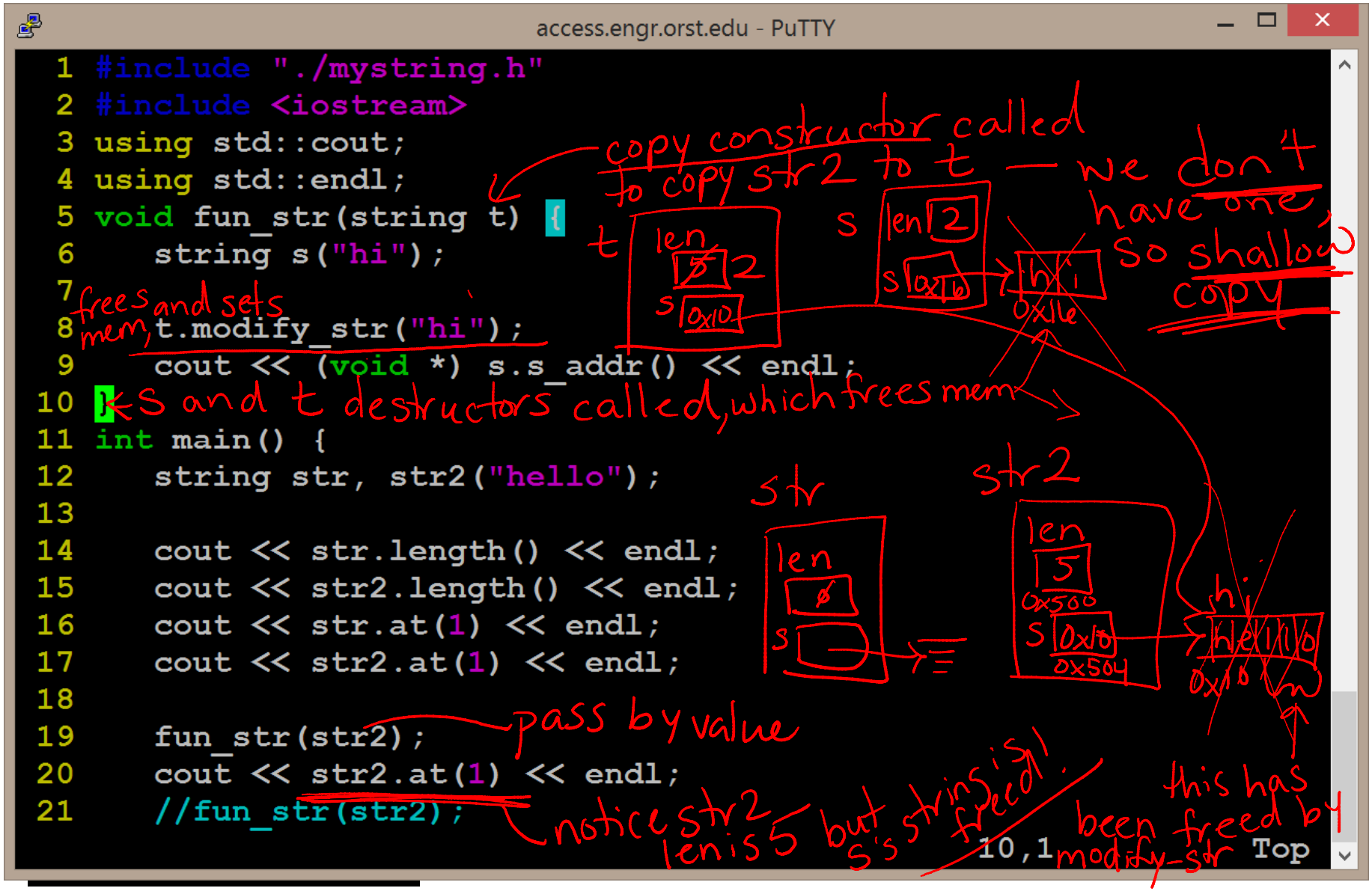
```
 1 #include "./mystring.h"
 2 #include <string.h> //same as cstring
 3 #include <iostream>
 4 using std::cout;
 5 using std::endl;
 6
 7 string::string() {  //should set s to NULL, and set len to zero
 8    s=NULL;
 9    len=0;
10 }
11 string::string(const char *str){
12    len=strlen(str);
13
14    s = new char[len];
15    for(int i=0; i<len; i++)
16       s[i]=str[i];
17 }
18 string::~string() {
19    delete [] s;
20    s=NULL;
21 }
"mystring.cpp" 59L, 1011C                    21,1          Top
```

default constructor, called when string str;

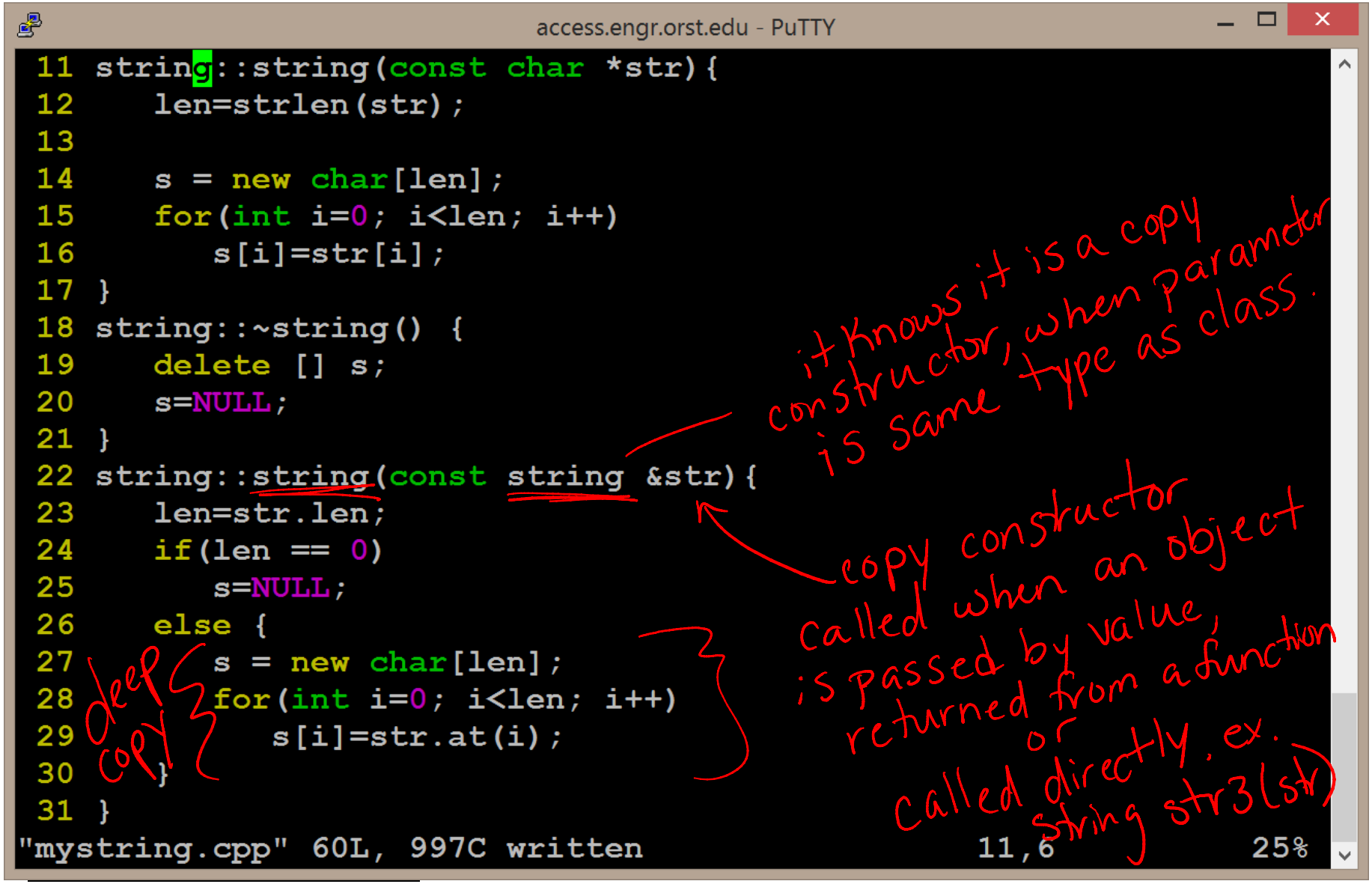
non-default constructor called when string str2("hello");

destructor to prevent memory leaks when object goes out of scope

Shallow Copy without Copy Constructor



mystring.cpp has destructor and copy constructor



Deep Copy with Copy Constructor

access.engr.orst.edu - PuTTY

```
1  #include "./mystring.h"
2  #include <iostream>
3  using std::cout;
4  using std::endl;
5  void fun_str(string t) {
6     string s("hi");
7
8     t.modify_str("hi");
9     cout << (void *) s.s_addr() << endl;
10 }
11 int main() {
12    string str, str2("hello");
13
14    cout << str.length() << endl;
15    cout << str2.length() << endl;
16    cout << str.at(1) << endl;
17    cout << str2.at(1) << endl;
18
19    fun_str(str2);
20    cout << str2.at(1) << endl;
21    //fun_str(str2);
```