

CS 162

Intro to CS II

Review: Classes



Assignment #1

- Why does our del example have three splats?

```
void del(int ***a);
```

- ① check that it is NULL before creating ~~*a = NULL;~~ we need to set pointer back to NULL!
- ② set it back to NULL, after deleting/freeing

- Why does is_valid_arguments have char *info[]?

```
is_valid_arguments(argv);
```

~~argv~~

could also pass by ref (int **&a)



```
1 #include <iostream>
2 #define M 4
3 #define N 6
4 using namespace std;
5
6 void del(int ***a){
7     for(int i=0; i<M; i++)
8         delete [] (*a)[i];
9     delete [] (*a);
10    *a = NULL;
11 }
12 void del(int **&a){
13     for(int i=0; i<M; i++)
14         delete [] a[i];
15     delete [] a;
16     a = NULL;
17 }
18 int main() {
19     int **a=NULL;
20     if(a==NULL) {
21         a = new int*[M];
22         for(int i=0; i<M; i++)
23             a[i]=new int[N];
24     }
25     del(a);
26     //del(&a);
27     return 0;
28 }
```

```
1 #ifndef MYSTRING_H
2 #define MYSTRING_H
3 class string{
4     private:
5         int len;
6         char *s;
7     public:
8         string(); //default should set s to NULL and len is zero
9         string(const char *); //set to specific string and change len
10        int length() const;
11 };
12
13 #endif
```

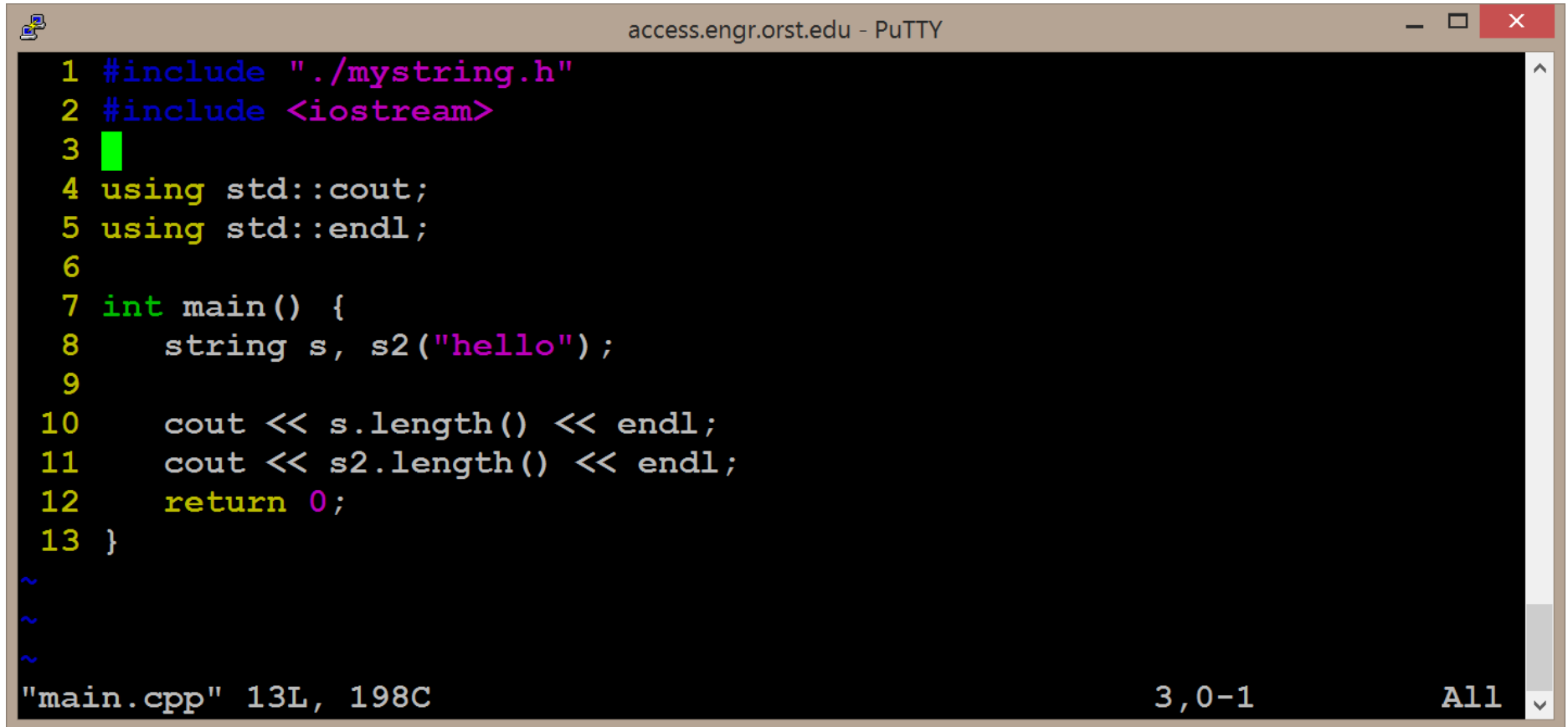
```
1 #include "./mystring.h"
2 #include <string.h> //or cstring for c-style strings
3
4 string::string() { //default should set s to NULL and len is zero
5     s=NULL;
6     len=0;
7 }
8 string::string(const char *str) { //set to str and change len
9     len=strlen(str);
10    s=new char[len];
11    for(int i=0; i<len; i++)
12        s[i]=str[i];
13 }
14 int string::length() const {
15     return len;
16 }
```

"mystring.cpp" 16L, 363C written

3,0-1

All

Test string class in mystring.h and .cpp



The image shows a PuTTY terminal window with a title bar that reads "access.engr.orst.edu - PuTTY". The terminal has a black background with syntax-highlighted C++ code. The code includes headers for a custom string class and the C++ standard library, uses cout and endl for output, and tests the length of two string objects in the main function. The status bar at the bottom of the terminal shows the filename "main.cpp", line 13, column 198, and other details.

```
1 #include "../mystring.h"
2 #include <iostream>
3
4 using std::cout;
5 using std::endl;
6
7 int main() {
8     string s, s2("hello");
9
10    cout << s.length() << endl;
11    cout << s2.length() << endl;
12    return 0;
13 }
```

~
~
~

"main.cpp" 13L, 198C 3,0-1 All

Test string class in mystring.h and .cpp



```
access.engr.orst.edu - PuTTY
flip1 ~/cs162/private/sec-2 171% g++ -c my
mystring.cpp  mystring.h      mystring.o
flip1 ~/cs162/private/sec-2 171% g++ -c mystring.cpp
flip1 ~/cs162/private/sec-2 172% g++ -c main.cpp
flip1 ~/cs162/private/sec-2 173% g++ my
mystring.cpp  mystring.h      mystring.o
flip1 ~/cs162/private/sec-2 173% g++ mystring.o main.o
flip1 ~/cs162/private/sec-2 174% a.out
0
5
flip1 ~/cs162/private/sec-2 175% █
```

What is const vs. static?

- What is const?
 - `const int x;` //cannot have as member var
 - `void function(const int &x) { ... }`
 - `void function() const { ... }`
 - When would we want to make a member function const? When wouldn't we?
- What is static?
 - Class variable or function
 - `static int x; Point::x`
- Can have a **static const int x=0;**