CS 162
Intro to CS II

Classes: Operator Overload and Friends

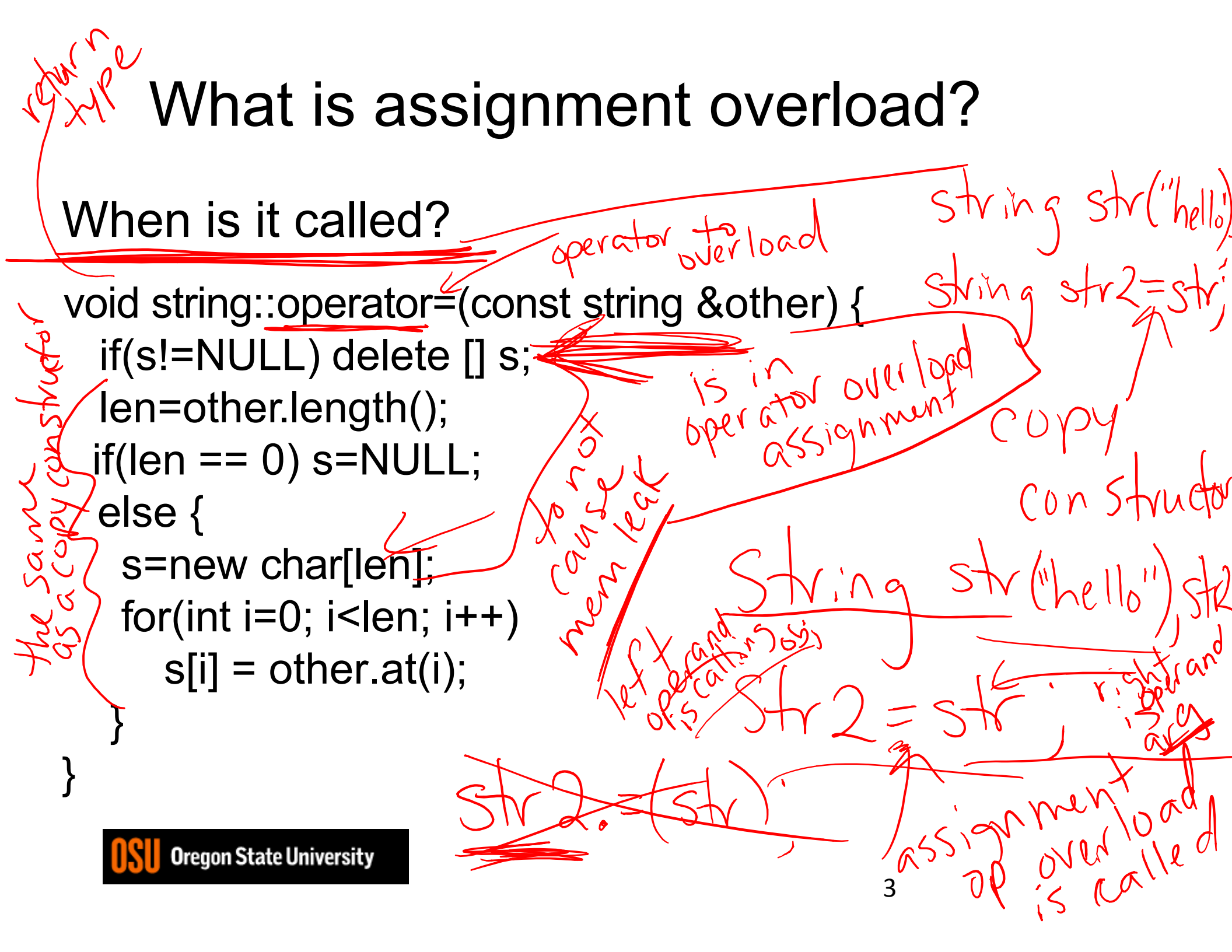
Revisit Big "Three"

- If dynamic memory allocation in class, then…
  - Destructor
  - Copy Constructor
  - Assignment operator overload

What is assignment overload?

When is it called?

```
void string::operator=(const string &other) {
  if(s!=NULL) delete [] s;
  len=other.length();
  if(len == 0) s=NULL;
  else {
    s=new char[len];
    for(int i=0; i<len; i++)
      s[i] = other.at(i);
  }
}
```



Operator Overload Non-Member

```
class Point {
public:
   Point();
   Point(int x_val, int y_val);    //Constructor
   void set_xy(int theX, int theY);    //Mutator Function
   int get_y();   //Accessor Function
   int get_x();   //Accessor Function
private:
   int x;
   int y;
};
int main () {
   Point p1, p2(2, 2);

   //How do we test if p1 is == p2?

   return 0;
}
Point::Point(int x_val, int y_val) {
     x=x_val;   y=y_val;
}
Point::Point() { x=0; y=0; }
```

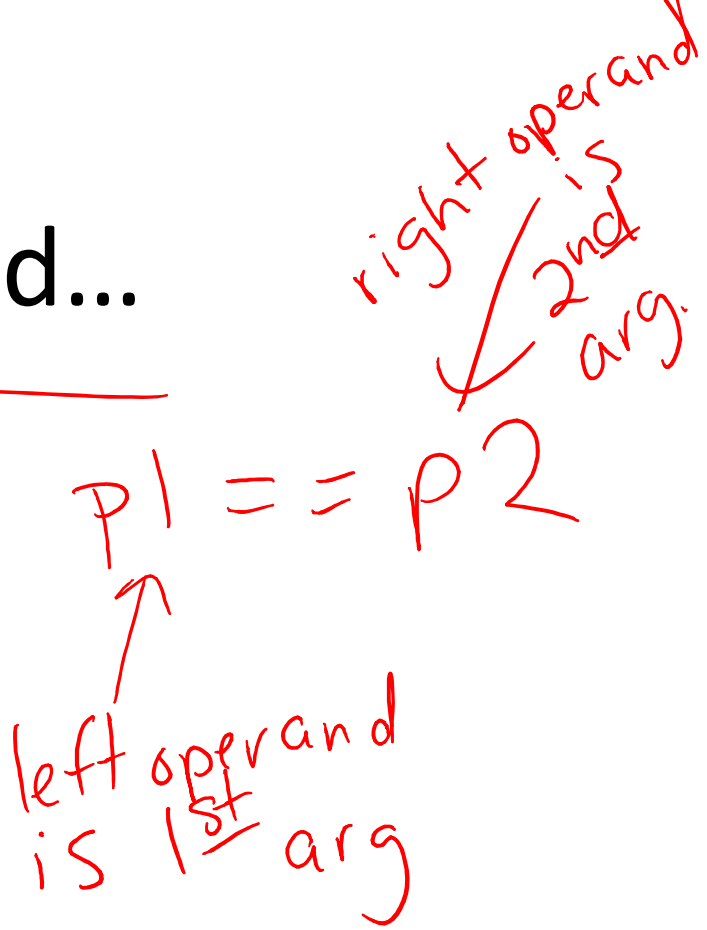
Operator Overload…

```
bool operator ==(const Point &p1, const Point &p2);
int main () {
  Point p1, p2(2, 2);

  //How do we test if(p1 == p2)?


  return 0;
}
bool operator ==(const Point &p1, const Point &p2) {
  if(p1.get_x() == p2.get_x() && p1.get_y() == p2.get_y())
    return true;
  else
    return false;
}
```

What if you don't want to go through the accessor/mutator functions?



Operator Overload Non-Member/Friends

```
class Point {
public:
   Point();
   Point(int x_val, int y_val);    //Constructor
   void set_xy(int theX, int theY);    //Mutator Function
   int get_y();    //Accessor Function
   int get_x();    //Accessor Function
   friend bool operator ==(const Point &p1, const Point &p2);
private:
   int x;
   int y;
};
Point::Point(int x_val, int y_val) {
     x=x_val;    y=y_val;
}
Point::Point() { x=0; y=0; }
```

Operator Overload/Friends…

```
bool operator ==(const Point &p1, const Point &p2);
int main () {
  Point p1, p2(2, 2);

  //How do we test if(p1 == p2)?


  return 0;
}
bool operator ==(const Point &p1, const Point &p2) {
  if(p1.x == p2.x && p1.y == p2.y)
    return true;
  else
    return false;
}
```

mystring.h

access.engr.orst.edu - PuTTY

```
 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
11       //Big Three
12       string(const string &); //copy constructor
13       ~string(); //destructor
14       void operator=(const string &); //assignment operator overload
15
16       friend bool operator==(const string &, const string &);
17       void set_s(const char *);
18       int length() const;
19       char at(int) const;
20       char * get_s() const;
21 };
22
23 #endif
                                                    17,31         All
```

mystring.cpp

access.engr.orst.edu - PuTTY

```
17 string::string(const string &str){
18    len=str.length();
19    if(len==0) s=NULL;
20    else {
21       s=new char[len];
22       for(int i=0; i<len; i++)
23          s[i]=str.at(i);
24    }
25 }
26 void string::operator=(const string &str){
27    if(s!=NULL) delete [] s;
28    len=str.length();
29    if(len==0) s=NULL;
30    else {
31       s=new char[len];
32       for(int i=0; i<len; i++)
33          s[i]=str.at(i);
34    }
35 }
36 string::~string(){
37    delete [] s;
38    s=NULL;
39 }
                                                    39,1          41%
```

Main.cpp

access.engr.orst.edu - PuTTY

```
 1 #include "./mystring.h"
 2 #include <iostream>
 3 #include <fstream>
 4
 5 using std::cout;
 6 using std::endl;
 7 using std::fstream;
 8 using std::ios;
 9 bool operator==(const string &s, const string &s2){
10    if(s.len==s2.len){
11       for(int i=0; i<s.len; i++)
12          if(s.s[i]!=s2.s[i])
13             return false;
14       return true;
15    }
16    else
17       return false;
18 }
19 int main() {
20    string s,s2("hello");
21    //string s=s2; //should call copy constructor because s is being made
22    s=s2; //calls the assignment overload because already constructed
23    cout << (s==s2) << endl; /
-- INSERT --                                              23,30         Top
```

OSU Oregon State University