ECE462
9/27/1989

OVERLOADING

f(string) — c1 = new Color(255, 255, 255);
f(Person) — c2 = new Color(0xFFFFFF);
f(int) — c3 = new Color(c3);

same name — Color(int, int, int);
Color(int)
Color(Color)

OPERATOR OVERLOADING (C++)

string s = "ECE";
string t = "462";
string u = s + t;

C++ allows user-defined operator overloading;
Java doesn't

copy constructor not called, unless & is removed from &arg1, &arg2

Operator Overloading

1. one operand must be a user-defined class
2. precedence can't be changed
3. can't change # of operands

IN Exam

```
Class MyComplex {
  public:
    MyComplex(double r, double i) { re = r, im = i }
    double getReal() const { return re; }
    double getImag() const { return im; }
  private:
    double re;
    double im;
}
```

```
int main() {
  MyComplex c1(2.5, 4.6);
  MyComplex c2(3.7, 1.9);
  MyComplex c3 = c1 + c2;
}
```

MyComplex c4 = c1; //calls CC

CC called twice if & is removed

try:

1. remove & in &arg1
2. c3(c1.getReal() + c2.getReal() + c1.getImag() + c2.getImag())

```
MyComplex operator+(const MyComplex &arg1, const MyComplex &arg2) {
  double r = arg1.getReal() + arg2.getReal();
  double i = arg1.getImag() + arg2.getImag();
  return MyComplex(r, i);
}
```