`cout << c1 << c2 << endl;`

~~ostream << operator~~

```
ostream & operator<< (ostream &os, const MyComplex &arg) {
  double r = arg.getReal();
  double i = ~~arg.getReal()~~; arg.getImag();

  os << d1 << " " << d2 << endl;
  return os;
}
```

OVERLOADING UNARY OPERATOR

```
MyComplex operator-(const MyComplex &arg) {
   return MyComplex(-arg.getReal(), -arg.getImag());
}

MyComplex c5 = -c1;
```

```
class MyComplex {
 public:
   MyComplex operator+ (const MyComplex &arg) {
      double r = re + arg.getReal();
      double i = im + arg.getImag();
      return MyComplex(r,i);
   }
};
```

```
MyComplex c3 = c1 + c2;
```

can't have both with operator+(MyComplex, MyComplex);
both equally perfect match

compiler doesn't automatically change operator → == doesn't ~~do~~ auto'd to !(!=)

friend ostream& operator<< (...);
enable access to private attribute;