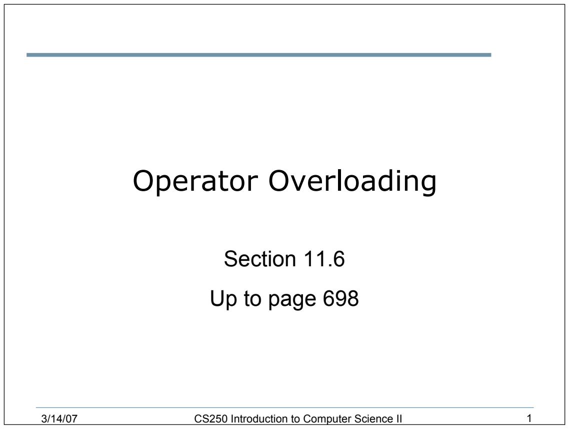
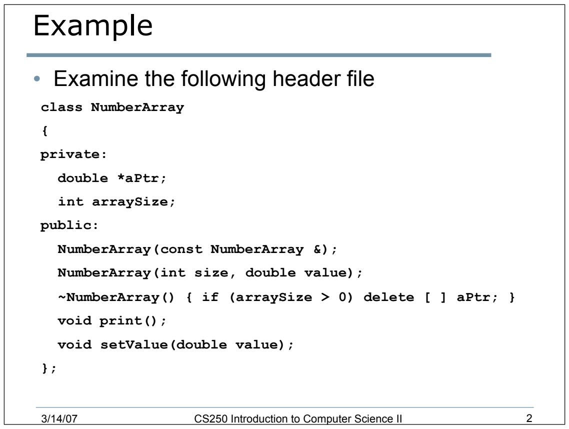


Operator Overloading

Section 11.6

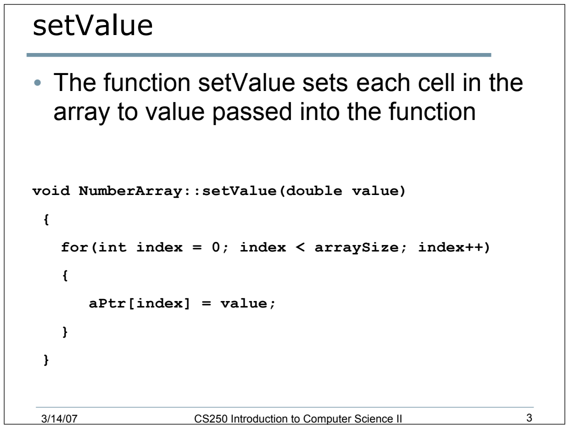
Up to page 698



Example

- Examine the following header file

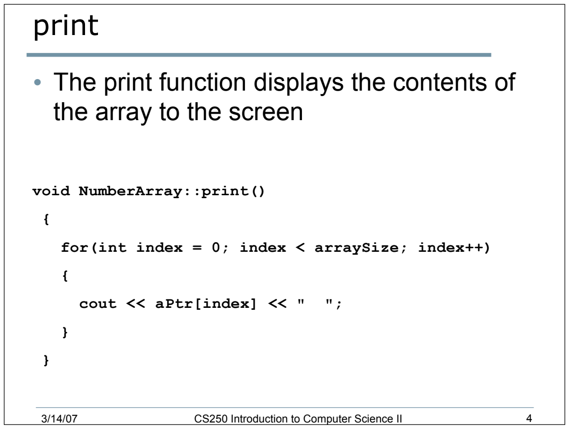
```
class NumberArray
{
private:
  double *aPtr;
  int arraySize;
public:
  NumberArray(const NumberArray &);
  NumberArray(int size, double value);
  ~NumberArray() { if (arraySize > 0) delete [ ] aPtr; }
  void print();
  void setValue(double value);
};
```



setValue

- The function setValue sets each cell in the array to value passed into the function

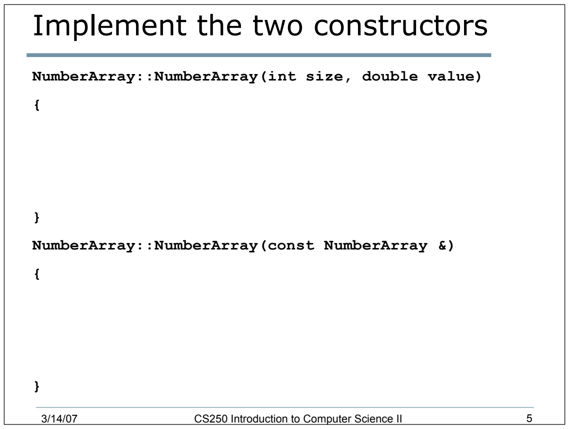
```
void NumberArray::setValue(double value)
 {
   for(int index = 0; index < arraySize; index++)
   {
      aPtr[index] = value;
   }
 }
```



print

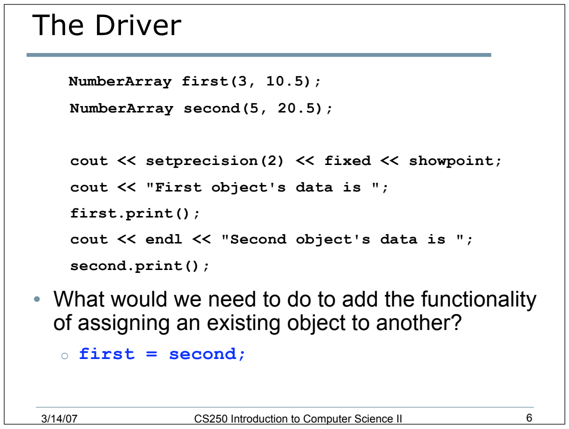
- The print function displays the contents of the array to the screen

```
void NumberArray::print()
 {
   for(int index = 0; index < arraySize; index++)
   {
     cout << aPtr[index] << "  ";
   }
 }
```



Implement the two constructors

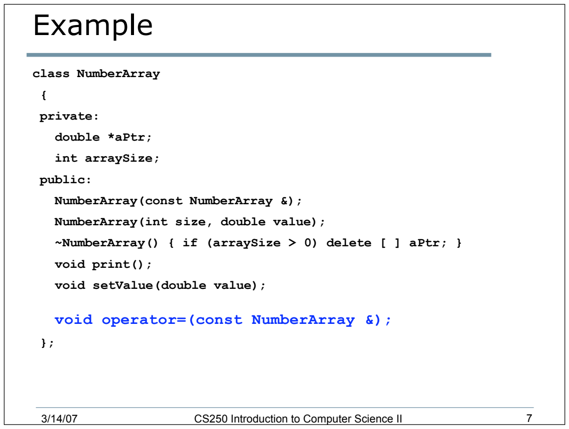
```
NumberArray::NumberArray(int size, double value)
{




}
NumberArray::NumberArray(const NumberArray &)
{




}
```



The Driver

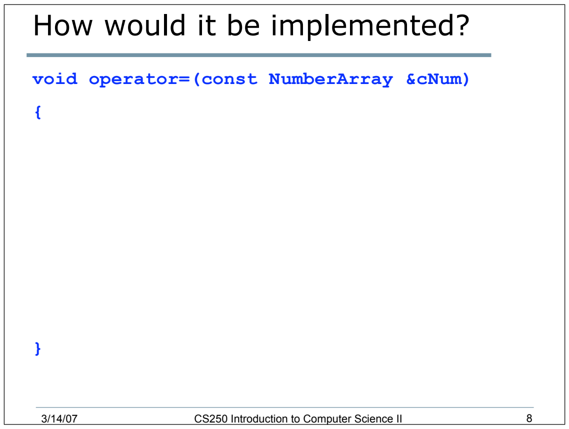
```
    NumberArray first(3, 10.5);
    NumberArray second(5, 20.5);

    cout << setprecision(2) << fixed << showpoint;
    cout << "First object's data is ";
    first.print();
    cout << endl << "Second object's data is ";
    second.print();
```

- What would we need to do to add the functionality of assigning an existing object to another?
  - `first = second;`



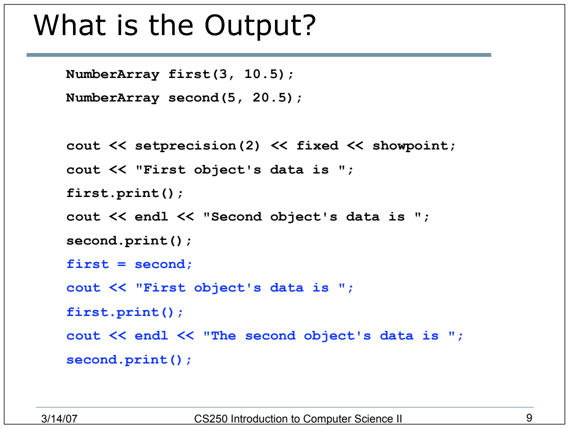
Example

```
class NumberArray
 {
 private:
   double *aPtr;
   int arraySize;
 public:
   NumberArray(const NumberArray &);
   NumberArray(int size, double value);
   ~NumberArray() { if (arraySize > 0) delete [ ] aPtr; }
   void print();
   void setValue(double value);

   void operator=(const NumberArray &);
 };
```



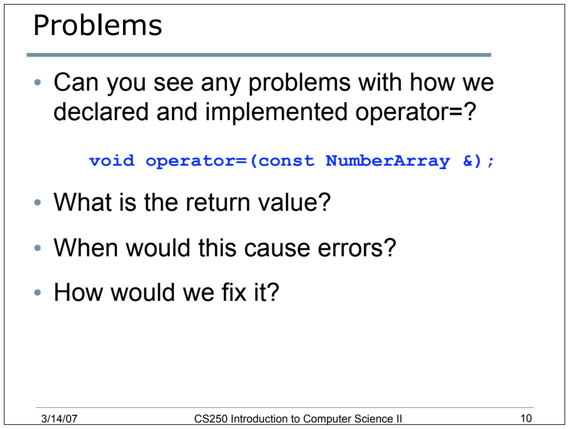
How would it be implemented?

```
void operator=(const NumberArray &cNum)
{







}
```



What is the Output?

```
    NumberArray first(3, 10.5);
    NumberArray second(5, 20.5);

    cout << setprecision(2) << fixed << showpoint;
    cout << "First object's data is ";
    first.print();
    cout << endl << "Second object's data is ";
    second.print();
    first = second;
    cout << "First object's data is ";
    first.print();
    cout << endl << "The second object's data is ";
    second.print();
```

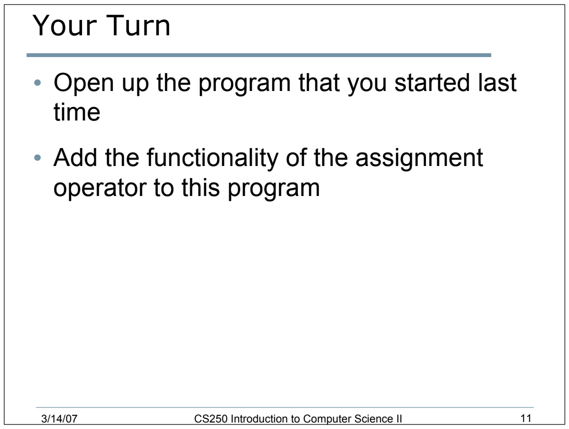


Problems

- Can you see any problems with how we declared and implemented operator=?

```
void operator=(const NumberArray &);
```

- What is the return value?
- When would this cause errors?
- How would we fix it?



Your Turn

- Open up the program that you started last time
- Add the functionality of the assignment operator to this program