

# CS 113 – Computer Science I

## Lecture 13 – Extending Arrays, Review

---

Adam Poliak

02/28/2023

# Announcements

- HW05 - loops
  - Due Tonight
  - Short implementing just 7 methods
- HW06 –
  - Due Friday after spring break
- Week after spring break
  - Hell week – I have no idea what to expect



# Agenda

- Announcements
- Array – our Bank
- Review

# Exercise: Nested loops

```
$ java Square
```

```
Enter a size: 5
```

```
*****
```

```
*****
```

```
*****
```

```
*****
```

```
*****
```

```
$ java Square
```

```
Enter a size: 1
```

```
*
```

```
$ java Square
```

```
Enter a size: 0
```

# Iterating through an array

Write a method called `printArray` that takes in an array of integers and prints out the values in each array:

*`printArray({1,2,3,4}) -> "1 2 3 4"`*

# Bank example

Keep track of account balances

Use an array:

- Each index represents another account

- The value represents the account's balance

Determine how many accounts we can hold:

- Create a new array of fixed size

# Bank

How can we find out how much money the bank is holding at once?

How can we find out which account is currently overdraft?

What other questions might the bank want to know?

# Bank example

Over time our bank becomes successful, lots of new clients

No more space for new customers

Implementation issue: running out of space in our array

Solution: build a bigger bank!



# Building a bigger bank



# Copying arrays

Old bank

3.0	6.0	7.0	-2.5
-----	-----	-----	------

# Copying arrays – build the new bank/array

Old bank

3.0	6.0	7.0	-2.5
-----	-----	-----	------

new bank

--	--	--	--	--	--

# Copying arrays – copy over values/customers

Old bank

3.0	6.0	7.0	-2.5
-----	-----	-----	------

new bank

--	--	--	--	--	--

# Copying arrays – copy over values/customers

Old bank

3.0	6.0	7.0	-2.5
-----	-----	-----	------



--	--	--	--	--	--

new bank

# Copying arrays – copy over values/customers

Old bank

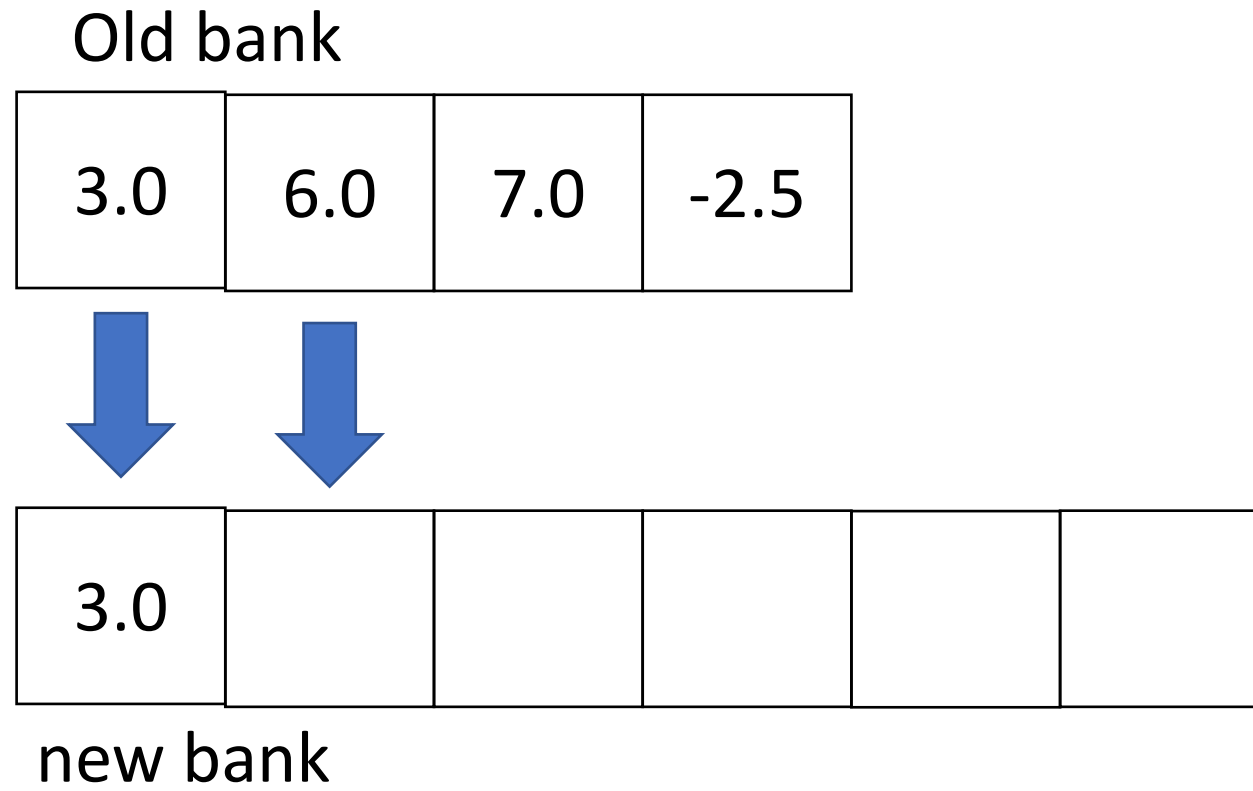
3.0	6.0	7.0	-2.5
-----	-----	-----	------



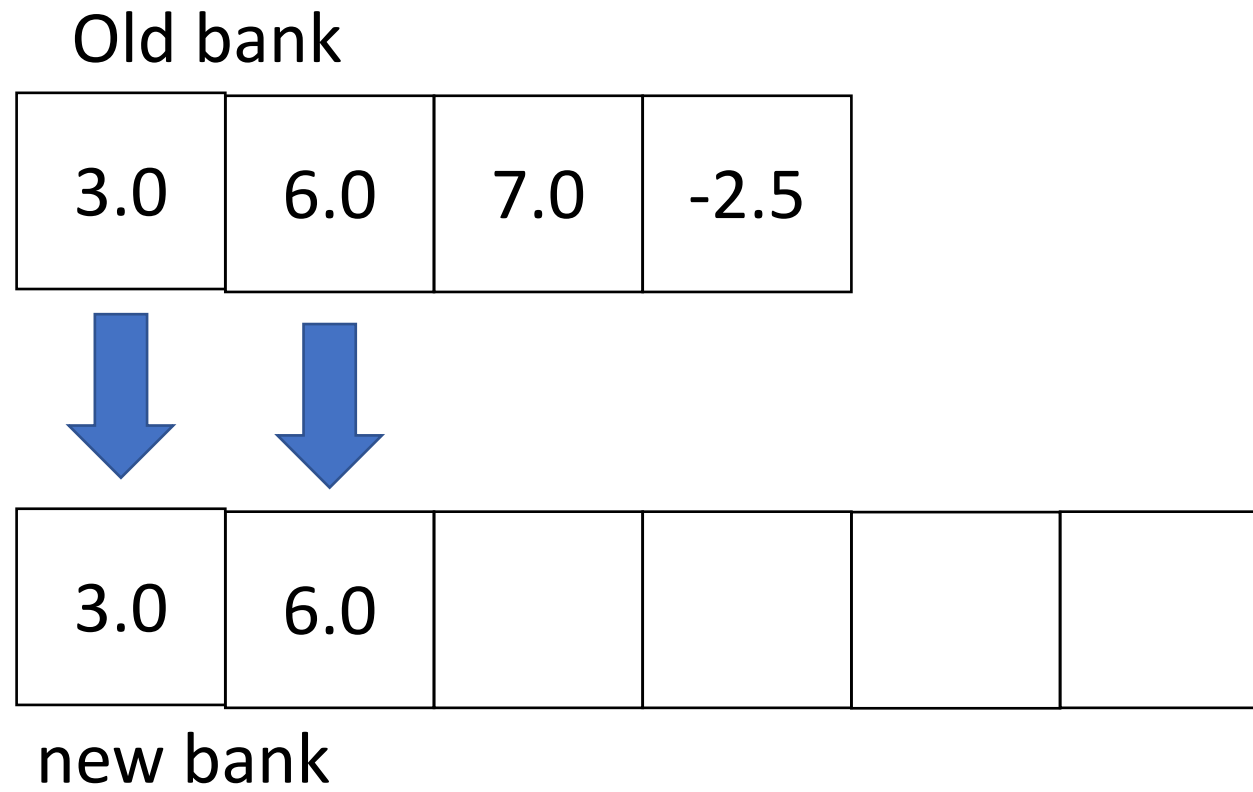
3.0					
-----	--	--	--	--	--

new bank

# Copying arrays – copy over values/customers

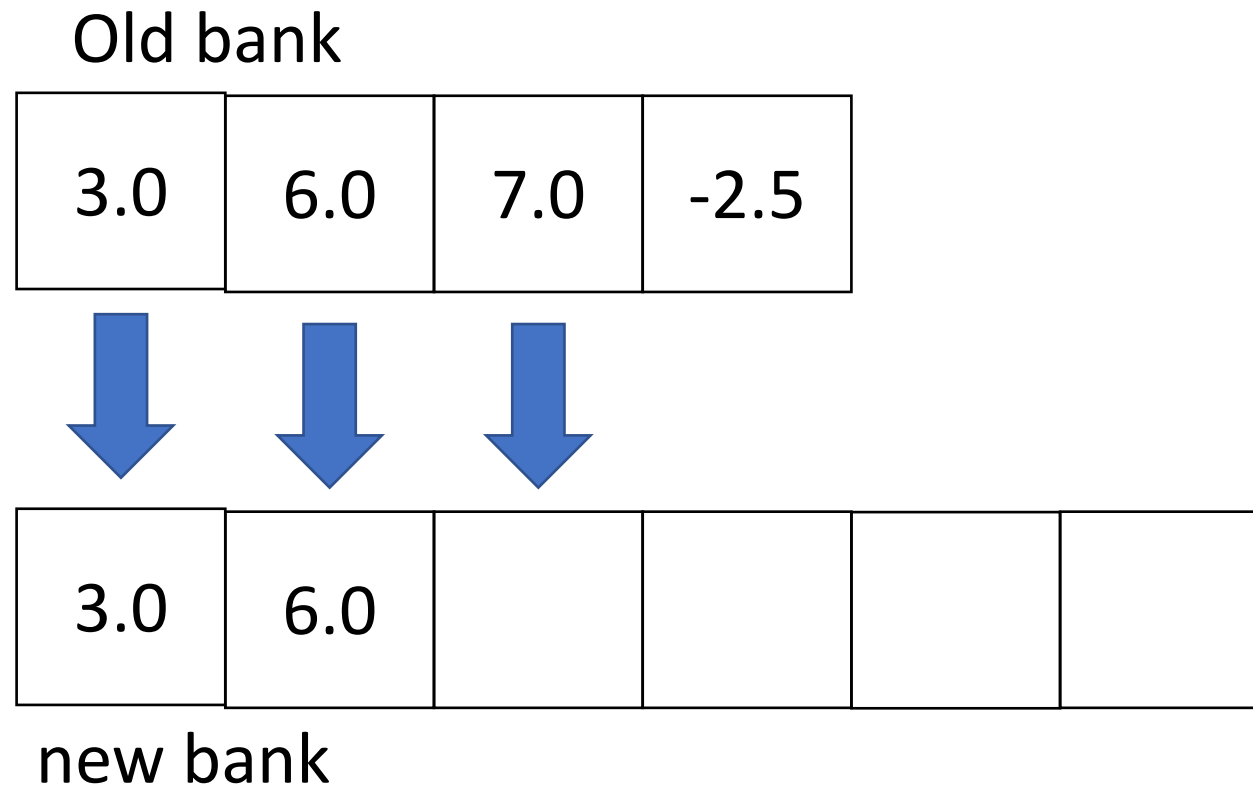


# Copying arrays – copy over values/customers

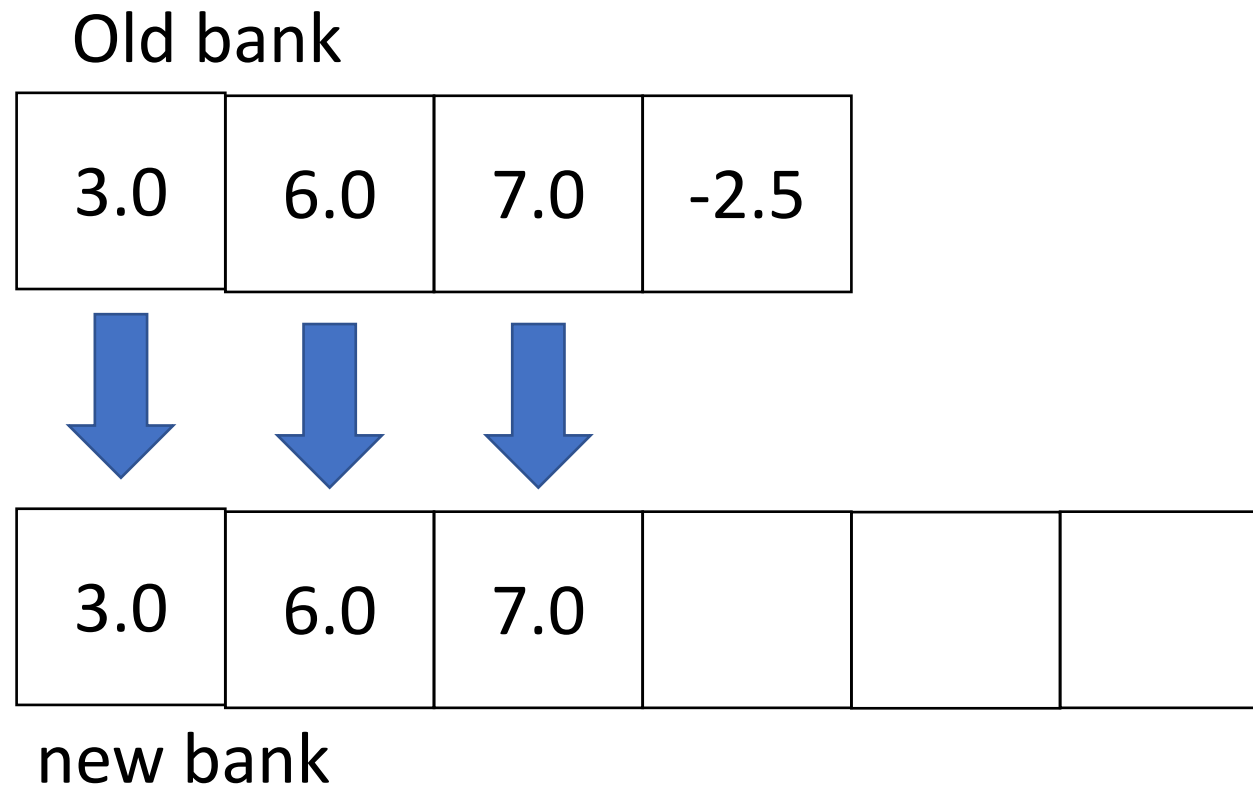




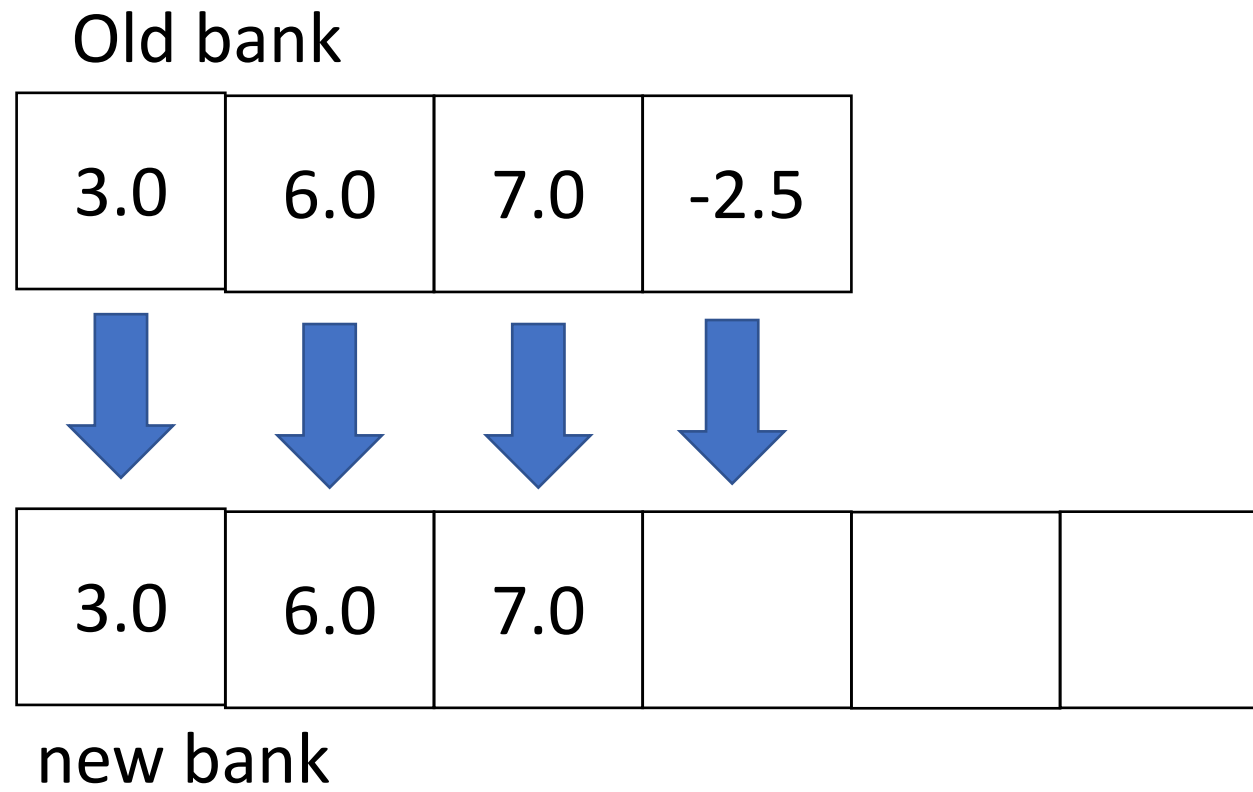
# Copying arrays – copy over values/customers



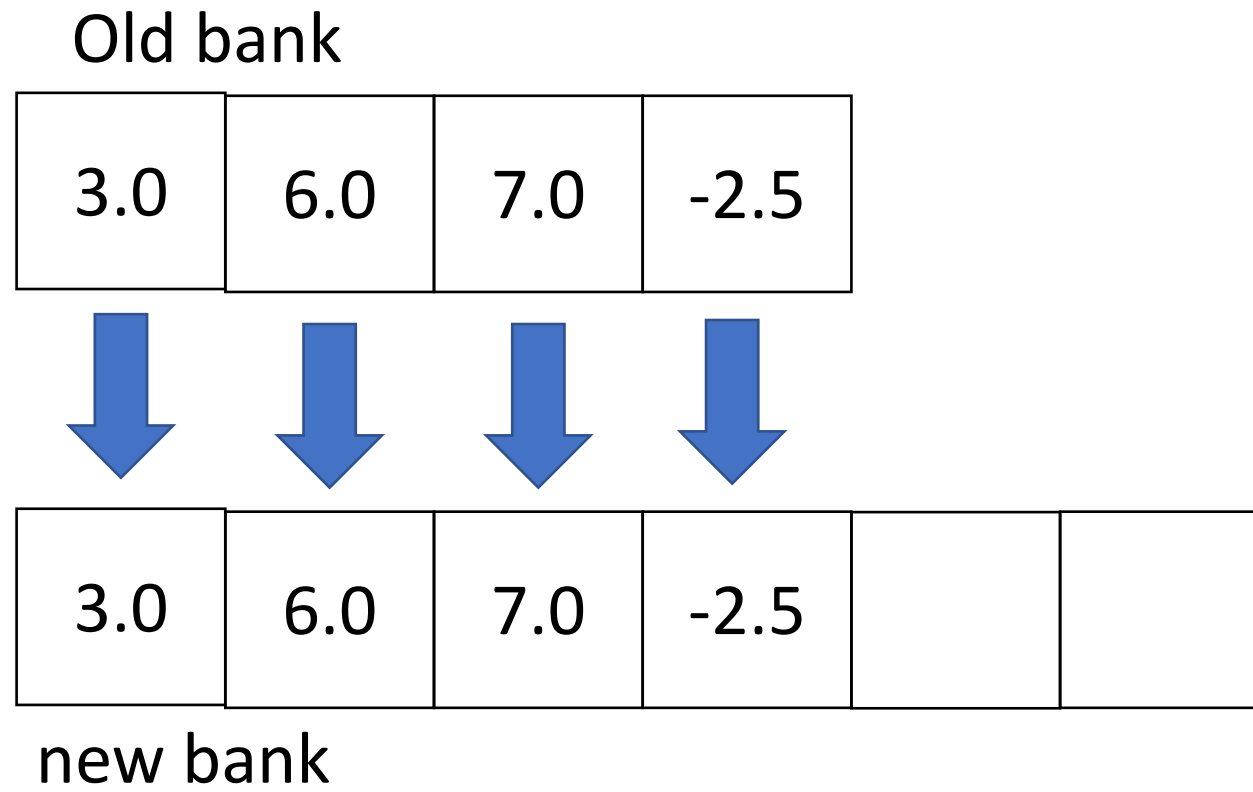
# Copying arrays – copy over values/customers



# Copying arrays – copy over values/customers



# Copying arrays – copy over values/customers



# Algorithm

When we run out of space in an array

- Create a new array (that's a bit bigger)
- Copy over all elements from the older array to the new array

How many steps do we take in this algorithm?

- Creating a new array – 1 step
- Copying  $n$  elements from the old array to the new array –  $n$  steps

# How big should the new array be?

Previous size plus 1

- Pro: not making too much space
- Con: might have to create new arrays a lot of times

As big as possible

- Pro: rarely have to create a new array
- Con: wasted space

Typical solution – previous size x 2



# Agenda

- Announcements
- Array – our Bank
- **Review**

# Midterm – Thursday 03/02

In class, closed book

Terminal commands, vim, directory structure  
variables (int, double, char, bool, string, array)

Expressions

Methods

Frame diagrams

Conditionals

Recursion

Loops

Practice exam is on course website



# Review

## Recursion

- When to make a recursive call vs when to stop

## Command line arguments