

# For Monday, 11/10

- No reading
- Recommended Practice Problems:
  - Chapter 10, problems 5-8
- Program 6 due
- Remember no class Friday

# Program 6

- Any questions?

# Creating Arrays

- How do we declare an array of Student objects?
- How do we create an array of Student objects?
- How do we put a Student object into the array?

# Accessing a Particular Element

# Processing the Whole Array

# Problem 1

- Write code to print all of the Students in our array, assuming that the Student class includes an appropriate toString method.

# Finding the One You Want

# Problem 2

- Write a method to locate and print a particular Student, given the Student's name. Assume that the Student class has a getName method.

# Primitive Array Declaration

- To specify that a variable is an array, we include square brackets, [], in the declaration.

- `int [] scores;`

- `char [] gradeArr;`

- The square brackets can come after the variable name, as they do in some other languages:

- `int scores[];`

- `char gradeArr[];`

# Review

- Since arrays are objects, we create them using what keyword?

# Array Creation

- In the creation, we have to specify the type and size of the array:

```
scores = new int[5];  
gradeArr = new char[10];  
price = new double[20];
```

- Once the array is created, the size of the array cannot be changed.

# Array Creation continued

- We often use **named constants** or variables for the size in an array declaration:

```
final int SIZE = 10;
```

```
final int MAX_ELEMS = 15;
```

```
int [] arr = new int[SIZE];
```

```
double[] flArr = new double[MAX_ELEMS];
```

# Accessing Individual Elements

- Subscripts (or indices) always start at 0, so an array with 5 elements has one at 0, one at 1, one at 2, one at 3, and one at 4.
- We access a particular array element by using the array name followed by the index in square brackets:  
    score[0]  
    arr[9]

# Using Array Elements

- All of the following are valid:  
score[0] = 4;  
score[0] += 7;  
score[1] = score[0] - 2;  
score[2] = score[1] + 5 \* score[0];  
score[j] = score[j + 1];
- Note: index can be any integral expression.

# Getting Data into Arrays

```
score[0] = 30;  
grade[3] = 'A';  
price[2] = 10.39;
```

# Array Initialization

- We can put initial values into an array when we create it.
- We must list all of the values:  
`int [] num = {58, 43, 60, 21, 38};`

# Array Practice

- Create an array to hold the tax for up to 10 different sales
- Create an array to hold the final letter grades for a class with up to 40 students
- Create an array of integers which holds the final average for those 40 students
- Create an array of characters with initial values 'a', 'd', 'y', and 'w'
- Assign  $TAX\_RATE * price$  to the first item in your first array

# Problem 1

- Write Java code to read values from the keyboard to fill the array `scores`. Input should stop when a negative number is entered. The maximum size of the array is in a constant `ARR_SIZE`.

# Problem 2

- Write Java code to add up the first `num_elements` values in the array `myVals` and store the sum in the variable `mySum`.