COMP 110
Introduction to Programming

Fall 2015
Time: TR 9:30 – 10:45
Room: AR 121 (Hanes Art Center)

Jay Aikat
FB 314, aikat@cs.unc.edu

Previous Class

• What did we discuss?
Today

• Announcements
  • Assignment 2: Due Friday, Oct 2 @ 11:55 PM
    http://comp110.com/assignments/the-worried-wizard

• Midterm on Thu, Oct 8
  • in class, no computers

• Study guide
  http://comp110.com/midterm-study-guide

• Arrays

Indices and For-Loops

• In programming, a for-loop usually starts with counter $i = 0$. There is a reason

```java
for (int i = 0; i < 5; i++) {
    scores[i] = keyboard.nextInt();
    scoreSum += scores[i];
}
```
Creating an Array

• Array is a special class and we create its objects
  – Syntax for creating an array:
    • `Base_Type[] Array_Name = new Base_Type[Length];`
  – Example:
    • `int[] pressure = new int[100];`
  – Alternatively:
    • `int[] pressure;`
    • `pressure = new int[100];`

Do not be OUT OF BOUNDS!

• Indices MUST be in bounds
  – `double[] entries = new double[5];` // from [0] to [4]
  – `entries[5] = 3.7;` // ERROR! Index out of bounds
• Your code will compile if you are using an index that is out of bounds, but it will give you a run-time error!
Initializing Arrays

• You can initialize arrays when you declare them
  – int[] scores = { 68, 97, 102 };
• Equivalent to
  – int[] scores = new int[3];
  – scores[0] = 68;
  – scores[1] = 97;
  – scores[2] = 102;
• Or, you can use for-loop
  – When in doubt, for-loop!

Joke

• Q: Why did the programmer quit his job?
• A: Because he didn't get arrays.
  Hint: A raise ;-)

COMP 110 - Fall 2015
2D Arrays

- Arrays having more than one index are often useful
  - Tables
  - Grids
  - Board games

<table>
<thead>
<tr>
<th></th>
<th>0: Open</th>
<th>1: High</th>
<th>2: Low</th>
<th>3: Close</th>
</tr>
</thead>
<tbody>
<tr>
<td>0: Apple Inc.</td>
<td>99.24</td>
<td>99.85</td>
<td>95.72</td>
<td>98.24</td>
</tr>
<tr>
<td>1: Walt Disney Co.</td>
<td>21.55</td>
<td>24.20</td>
<td>21.41</td>
<td>23.36</td>
</tr>
<tr>
<td>2: Google Inc.</td>
<td>333.12</td>
<td>341.15</td>
<td>325.33</td>
<td>331.14</td>
</tr>
<tr>
<td>3: Microsoft Corp.</td>
<td>21.32</td>
<td>21.54</td>
<td>21.00</td>
<td>21.50</td>
</tr>
</tbody>
</table>

Declaring and Creating 2D Arrays

- Two pairs of square brackets means 2D
  - `int[][] table = new int[3][4];`
- or
  - `int[][] table;`
  - `table = new int[3][4];`
Declaring and Creating 2D Arrays

- Array (or 1D array) gives you a list of variables
  - int[] score = new int[5] gives you score[0], score[1], ..., score[5]
- 2D array gives you a table of variables
  - int[][] table = new int[3][4];

Using a 2D Array

- We use a loop to access 1D arrays

```java
for (int i = 0; i < 5; i++) {
    scores[i] = keyboard.nextInt();
    scoreSum += scores[i];
}
```
Using a 2D Array

- We use nested loops for 2D arrays

```java
int[][] table = new int[4][3];
for (int i = 0; i < 4; i++) {
    for (int j = 0; j < 3; j++) {
        table[i][j] = i * 3 + j;
        System.out.println(table[i][j]);
    }
}
```

Multidimensional Arrays

- You can have more than two dimensions
  - `int[][][] cube = new int[4][3][4];`
- Use more nested loops to access all elements
  - for (int i...)
    - for (int j...)
      - for (int k...)
Arrays Example

```java
import java.util.*;
public class SampleArray {
    public static void main(String[] args) {
        Scanner keyboard = new Scanner(System.in);
        int[] myArray = new int[5];

        for (int i = 0; i < myArray.length; i++) {
            System.out.println("Please input a number");
            myArray[i] = keyboard.nextInt();
            System.out.println(myArray[i]);
        }
        System.out.println(Arrays.toString(myArray));  // to print the entire array
    }
}
```

Comparing Scores/Averages w/ Arrays

```java
System.out.println("Enter 5 basketball scores:");
Scanner keyboard = new Scanner(System.in);
int[] scores = new int[5];
int scoreSum = 0;
for (int i = 0; i < 5; i++) {
    scores[i] = keyboard.nextInt();
    scoreSum += scores[i];
}

double average = (double) scoreSum / 5;
System.out.println("Average score: "+ average);
for (int i = 0; i < 5; i++) {
    if (scores[i] > average)
        System.out.println(scores[i] + ": above average");
    else if (scores[i] < average)
        System.out.println(scores[i] + ": below average");
    else
        System.out.println(scores[i] + ": equal to the average");
}
```
Next class

• More on arrays