Arrays

Oct 9

making and using arrays, pointers

Midterm 1

Average: 104
Median : 103

```
1         int n = 5;
2         for (int i = 0; i <= n; i++) {
3             for (int j = n; j > i; j = j - 1) {
4                 if ((i + j) >=7) {
5                     System.out.println("big");
6                 } else if ((i+j) > 4) {
7                     System.out.println("medium");
8                 }
9             }
10         }
```

Why Arrays

- Consider a variable
  - Holds a single value
    ```
    int x =8;
    ```
- Often have programs deal with a group of values
  - want to be able to handle them with code that takes advantage of their grouping
    - Use loops to do things with all of the group

- Answer: arrays

Making and Using Arrays

in Java

- Make:
  - An array of a known length:
    ```
    int[] intArray = new int[4];
    ```
  - An array of a length known at runtime:
    ```
    int number = 7;
    double[] doubleArray = new double[number];
    ```
- Using arrays:
  - Arrays know how big they are:
    ```
    doubleArray.length==7 // true!!
    ```
  - give name and location
    - location is "zero indexed"
      ```
      System.out.println(intArray[0] + "  " + doubleArray[1]);
      ```
  - Locations must be:
    - non-negative
    - less than the length of the array

picture of what actually happens as a result of doing this, including pointers, memory allocation, data type, []

Arrays setting / getting

- Arrays define things that are just like other variables
  - `aa[4]` is not much different than any other variable
  - Set it: aa[4] = 17;
  - Read it: System.out.println(aa[4]);
- The important thing about arrays:
  - They make it explicit that things are related
  - You can use loops on them.

Arrays

and loops

```
public class AandL {
    public static void main(String[] args) {
        int[] nums = new int[5];
        for (int i = 0; i < nums.length; i++) {
            nums[i] = i * 2;
        }
        for (int i = 0; i < nums.length; i++) {
            System.out.println(nums[i]);
        }
    }
}
```

why not <= ?

Create Array

Fill with multiples of 2
Suppose I wanted powers of 2?

Print contents
NOT System.out.println(nums);

i,j,k traditionally used to index array
This is only case for single letter variable names

Finding the max in an array

Checking Random Number Generation

With Arrays!

Idea. Draw 1000 random number in the range 0..XX.
if good generator, then frequency of each number should be similar (ish)

Algorithm???

Arrays -- Example

Test Grades

- Suppose you have a class with three people: Mary, Fran and Yen
- You want to compute the average grade

```
public class NoArray {
    public static void main(String[] args) {
        int mary = 5;
        int fran = 6;
        int yen = 10;
        int sum = 0;
        sum += mary;
        sum += fran;
        sum += yen;
        double average = ((double) sum) / 3;
        System.out.println(average);
    }
}
```

```
public class WithArray {
    public static void main(String[] args) {
        int[] grades = new int[3];
        grades[0] = 5;
        grades[1] = 6;
        grades[2] = 10;
        int sum = 0;
        for (int i = 0; i < grades.length; i++) {
            sum += grades[i];
        }
        double average = ((double) sum) / grades.length;
        System.out.println(average);
    }
}
```

Now suppose the class has 30 people rather than 3.

Chalkboards

- Expand the grades program
  - 5 people
  - Store their names in an array as well as their grades. Hint, you are going to need another array
  - Print a table with columns name and grade
  - Finish by printing average

```java
public class WithArray {
    public static void main(String[] args) {
        int[] grades = new int[3];
        grades[0] = 5;
        grades[1] = 6;
        grades[2] = 10;
        int sum = 0;
        for (int i = 0; i < grades.length; i++) {
            sum += grades[i];
        }
        double average = ((double) sum) / grades.length;
        System.out.println(average);
    }
}
```