Arrays
Oct 9

making and using arrays, pointers
Midterm 1

Average: 104
Median: 103

```java
int n = 5;
for (int i = 0; i <= n; i++) {
    for (int j = n; j > i; j = j - 1) {
        if ((i + j) >= 7) {
            System.out.println("big");
        } else if ((i+j) > 4) {
            System.out.println("medium");
        }
    }
}
```
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:
    ```java
    int[] intArray = new int[4];
    ```
  • An array of a length known at runtime:
    ```java
    int number = 7;
    double[] doubleArray = new double[number];
    ```

• Using arrays:
  • Arrays know how big they are:
    ```java
    doubleArray.length==7 // true!!
    ```
  • give name and location
    • location is "zero indexed"
      ```java
      System.out.println(intArray[0] + "  " + doubleArray[1]);
      ```
  • Locations must be:
    • non-negative
    • less than the length of the array
Arrays setting / getting

- Arrays define things that are just like other variables
  - `aa[4]` is not much different than any other variable
    - 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.
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 <= ?

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

Create Array

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

```java
class Maxx {
  public static void main(String[] args) {
    // step one create array and fill it
    double[] dArray = new double[10];
    for (int i = 0; i < dArray.length; i++) {
      dArray[i] = Math.random();
    }
    // Now get the max
    double maxx = 0.0;
    // this is a bad way to start
    for (int i = 0; i < dArray.length; i++) {
      if (maxx < dArray[i]) {
        maxx = dArray[i];
      }
    }
    // Finally print the array and the max
    for (int i = 0; i < dArray.length; i++) {
      System.out.println(i + "  " + dArray[i]);
    }
    System.out.println("The max of the above is " + maxx);
  }
}
```
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

```java
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);
    }
}
```

```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);
    }
}
```

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);
    }
}
```