ITI 1120
Lab #5

Arrays

Contributors: G. Arbez, M. Eid, Sylvia Boyd, Romelia Plesa, D. Inkpen, A. Williams, D. Amyot

Lab Agenda

- Arrays
  - Declaring reference variables to arrays
  - Creating arrays
  - Exercises
- Formatting numerical results

Arrays in Java

- Declarations of reference variables to arrays.

  GIVEN: anArray *(Reference to an array of real values)*

  ```
  double[] anArray; // Reference
  ```

- Creating the array:

  D ← makeNewArray(5) (length 5)

  ```
  d = new double[5];
  ```

Array creation

- Until the array has been created, the reference of the array contains the special value **null** (actually is the address 0)
  - Attempting to use a reference variable containing null (for example, d[0]), will result in a "null pointer exception" that will crash a program.
- After the array has been created (with new), the elements in the array have not been assigned any value (in Java, they normally have the value 0).
  - Trying to access an element with an index outside the allowed range (ex. d[5]) will crash the program (*Array index out of bounds exception)*

Array lengths

- You can ask an array for its length:

```
double[] d;
int len;
d = new double[4];
len = d.length;
// len is now 4
```

- Notes:
  - Do not use `[]` or `()` when asking for the length
  - Attempting to ask for the length of an array before creating it will result in a "null pointer exception" that will crash the program.
  - The length is NOT the maximum index - it is one larger

Reading arrays from the keyboard

- Use the provided **`ITI1120`** class (get it from the virtual campus, Lab area):

`ITI1120.readIntLine()`

- Reads an array of integers
- The result is of type **`int[]`** **(a reference to an array)**
- **You may check the array length (using `length`) if necessary.**

`ITI1120.readDoubleLine()`

- Similar to **`readIntLine()`**, except for **`double`** values

`ITI1120.readCharLine()`

- Reads an array of character values
- Includes ALL characters, including spaces

Examples of using ITI1120

```
int x = ITI1120.readInt( );
```

- If you type 123 and hit ENTER, x will be assigned the value 123.

```
int[] x; // declares a reference variable
x = ITI1520.readIntLine( );
```

- If you type 12 3 0 -546 (numbers are separated by **spaces**) and hit ENTER, x will be assigned a reference to an array with elements, where x[0] contains 12, x[1] contains 3, x[2] contains 0, x[3] contains -546 .
- The array is created by readIntLine (using new) al so there is not need to create the array explicitely (but the reference variable must be declared to receive the reference)
- The methods readDouble and readDoubleLine work the same way.
- The method readCharLine places ALL characters (including spaces) in a char array.

Exercise 1: Calculate the Average

- Complete the algorithm that will calculate the average of an array of numbers of length n. Start with Lab5Ex1.doc.
- Implement your algorithms in Java as methods
  - Develop a main algorithm/method to obtain an array of values from the user, call the algorithm/method to compute the average, and print the results.

Exercise 2 – Computing Statistics

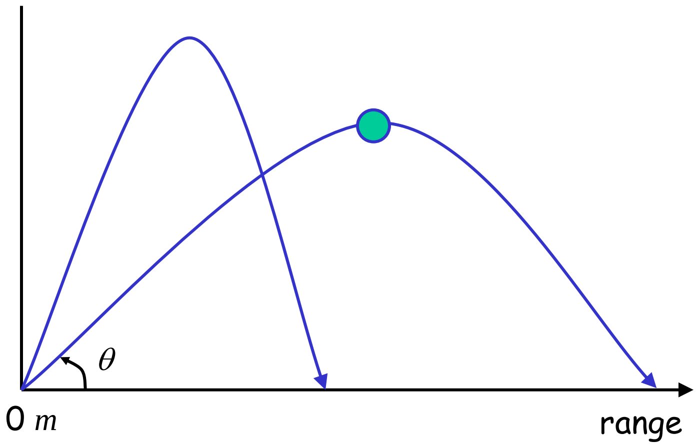
- Write a program that inputs a number of student grades and finds: the average the marks, the maximum grade, and the minimum grade.
  - Develop your algorithms with Visio and Word – start with Lab5Ex2.doc.
  - Translate the algorithms to Java. Start with Template.java (you can also start with Lab5Ex1.java).
  - The main algorithm method gets the grades from the students (stored in an array) calls the statistics() algorithm/method to compute the average, maximum, and minimum. It also prints the results to the user.
  - The statistics algorithm/method receives a reference to the array and the size of the array as parameters and computes the average, minimum, and maximum.
    - Note that in the case of the algorithm, a reference to the array and number of elements are the algorithm results. In the case of Java, only a single value is returned – the reference to the array.

Exercise 3: Throwing a Ball

- Write an algorithm that will calculate how far a ball thrown at v metres per second will travel horizontally in metres, depending on the angle θ (in degrees) at which it is thrown.
- Return an array of values where:

  range[0]: ball is thrown at 0 degrees above horizontal
  range[1]: ball is thrown at 10 degrees above horizontal
  …
  range[9]: ball is thrown at 90 degrees above horizontal
  (that is, straight up).
- Complete your program with a main algorithm that gets a value for speed from the user, calls the algorithm to generate the array described above, and prints the contents of the array.
- Develop your algorithms with Visio and Word – start with Lab5Ex3.doc.

Exercise 3: Throwing a Ball



Exercise 3: Throwing a Ball

- Formula:

$$range = \frac{2v^2 \cos\theta_r \sin\theta_r}{g}$$

θ_r = angle in radians

- Where:

$$\theta_r = \frac{\pi}{180}\theta$$

g = 9.8, constant acceleration due to gravity.

Exercise 3: Throwing a Ball

- Implement your algorithms in Java. Start with Template.java
- Useful items:
  - **`Math.PI`**: value of π.
  - **`Math.sin(x)`**: sine of **`x`**, where **`x`** is in radians
  - **`Math.cos(x)`**: cosine of **`x`**, where **`x`** is in radians

Output formatting

- Java provides several facilities for output formatting.
  - Decimal formatter: formats numeric values
- The specific results of output formatting will depend on the "Regional settings" in the control panel of your computer.
  - Setting the regional settings to "French(Canada)" will result in different formats for:
    - currency ($ at end)
    - decimal points and thousands separators are switched.

The decimal formatter

- Uses a "format string" to identify the locations of various types of characters.

#	Location of optional digit; if no digit, no value is printed
0	Location of non-optional digit; if no digit, a zero is printed
.	Location of decimal point
,	Location of thousands separators
E	Location of exponent indicator in powers of 10
$	Location of currency symbol

Example

- Results for a `double` value of `12345.6`

########	12346	Note rounding, left justification
00000000	00012346	Extra 0 digits on left
######.##	12345.6	First, last digits not used
######.00	12345.60	Forces 2 decimal places
###,###.00	12,345.60	Uses comma separators
$###,###.00	$12,345.60	Note that $ is moved to be next to first digit printed
#.###E00	1.235E04	Note rounding, 2 digits of exponent

Using a Decimal formatter

There is a two step process to format numbers:

- Create a new decimal formatter, and store it in a variable of type DecimalFormat

```
DecimalFormat df = new DecimalFormat("#####.##");
```

- Use it for each value that is to get the same format.

```
System.out.println("Value1 = " + df.format( value1 ) );
```

Decimal formatter – java e.g.

```
import java.text.DecimalFormat;  // used to find DecimalFormat class

public class FormatTest
{
   public static void main(String[] args)
   {
      DecimalFormat df = new DecimalFormat("#####.##");

      double value1;
      double value2;

      value1 = 12345.6;
      value2 = 34.5678;
      System.out.println("Value1 = " + df.format( value1 ) );
      System.out.println("Value2 = " + df.format( value2 ) );
   }
}
```

Experiments to try with formatting

- What happens if you try to format an integer with more digits that provided for in the format?
  - example: format 1234 with ###.
- What happens if you include a % character at the end of the format?
- How would you print a set of double values so that the decimal points line up?
  - Example:

```
1234.56      1234.56
0234.56       234.56
0034.56        34.56
(easier)       (hard)
```

Supplemental Exercise: Standard Deviation

- Try this exercise at home.

- The standard deviation of a set of values is a measure used in statistics to provide information about how much a set of values diverge from the average. That is, the standard deviation gives us a sense of how far a "typical" value is away from the average.

- For example, the average of all grades in a course could be 73 out of 100. In the unlikely case that everyone received a grade of 73, the standard deviation would be zero. In a more typical set of grades, the standard deviation could be a value such as 13.75.

Standard Deviation

- Suppose that you have n data values, and these data values are represented by $\{x_i\}$ where $0 \le i < n$.
- The standard deviation s is calculated using the following formula:

$$s = \sqrt{\frac{(x_0 - a)^2 + (x_1 - a)^2 + \cdots + (x_{n-1} - a)^2}{n-1}}$$

where

$$a = \frac{x_0 + x_1 + \cdots + x_{n-1}}{n}$$

is the average of the data values.

Software: Standard Deviation

- Develop an algorithm that calculates standard deviation of the values in an array referenced by X and has size N. Use Lab5ExSupplemental.doc. The main algorithm is provided.

- Implement both the main and problem solving algorithms in Java