# CST8130: Data Structures Lab #1- Using Arrays

***DUE: demonstrate in lab during week January 15 – January 21 (or earlier)***

## Problem Description:

Use the Lab1 – Using Arrays download available with this lab. It contains two classes – ***Numbers*** and ***Lab1Main.*** You should load these classes into a project, and write the Java code as indicated in the classes according to the following specifications. NOTE: your code must never, never, ever “crap out”. (You must handle every possible condition). Test your code through the Lab 1 test plan.

The purpose of this lab is to understand object references, array allocation, memory allocation and constructors in Java.

**Class Numbers**

Data members:

* a reference to a dynamically allocated array of ***Float*** references
* an ***int*** to hold the size of the array
* an *int* to hold the number of values currently in the array

Methods

* default constructor
* initial constructor using an ***int*** parameter to set size of the array
* method ***initValuesinArray*** – which will prompt the user to enter float values to fill the array.
* method ***calcAverage***– will return a ***float*** which is a average of the values in the array
* method ***toString*** – will return a ***String*** of the values in the array

**Class Lab1Main**

Data members:

* none

Methods

* ***main*** – a menu which creates an object of type ***Numbers***, and prompts users to execute each of the methods above (and prompts for input from user when needed ..ie for parameter values to send to methods). See sample output for details

**Sample Output: (green is user input)**

Enter 1 to initialize a default array,

2 to initialize an array of input size,

3 fill array with values,

4 display values in array,

5 to display average of the values in the array,

6 to quit

3

Enter the float numbers as values in the array:

Enter value: 1

Enter value: 2

Enter value: 3

Enter value: 4

Enter value: 5

Enter value: 6

Enter value: 7

Enter value: 8

Enter value: 9

Enter value: 10

Enter 1 to initialize a default array,

2 to initialize an array of input size,

3 fill array with values,

4 display values in array,

5 to display average of the values in the array,

6 to quit

4

Values are:

1.0

2.0

3.0

4.0

5.0

6.0

7.0

8.0

9.0

10.0

Enter 1 to initialize a default array,

2 to initialize an array of input size,

3 fill array with values,

4 display values in array,

5 to display average of the values in the array,

6 to quit

5

The average is 5.5

Enter 1 to initialize a default array,

2 to initialize an array of input size,

3 fill array with values,

4 display values in array,

5 to display average of the values in the array,

6 to quit

2

Enter new size of array: 3

Enter 1 to initialize a default array,

2 to initialize an array of input size,

3 fill array with values,

4 display values in array,

5 to display average of the values in the array,

6 to quit

3

Enter the float numbers as values in the array:

Enter value: 10

Enter value: 11

Enter value: 12

Enter 1 to initialize a default array,

2 to initialize an array of input size,

3 fill array with values,

4 display values in array,

5 to display average of the values in the array,

6 to quit

5

The average is 11.0

Enter 1 to initialize a default array,

2 to initialize an array of input size,

3 fill array with values,

4 display values in array,

5 to display average of the values in the array,

6 to quit

4

Values are:

10.0

11.0

12.0

Enter 1 to initialize a default array,

2 to initialize an array of input size,

3 fill array with values,

4 display values in array,

5 to display average of the values in the array,

6 to quit

7

Invalid entry...

**In preparation for next week's lecture:**

Note what happens when you run this program with size 100, 10000, 1000000, 3000000000….