

### Lab 07 – Loops; One-Dimensional Arrays

#### Introduction

This lab assignment continues to give you practice using loops, particularly loops with variable termination conditions, and it also provides you an opportunity to use one-dimensional arrays.

Recall that an array is used to store a collection of data. The data can be values of Java primitive data types or else objects (for instance, **String** objects), but the data stored must all be of the same type. An array is created in Java using a statement of the form

$$\text{BaseType}[] \text{ arrayName} = \text{new BaseType}[\text{length}];$$

where *BaseType* is the underlying type of data stored in the array, *arrayName* is just a variable name used to refer to the array, and *length* is a positive integer indicating the number of elements in the array. An array element can be referred to using *arrayName*[*i*], where *i* is an integer index (starting with 0). The length of the array is stored in the variable *arrayName*.length.

The arrays used in this assignment will all be one-dimensional (that is, they won't be arrays of arrays), and they will all use Java primitive data types as the base type.

For the assignment, you are to create a program using two arrays of type **double** to store multiple values for *x* and *y*, where *y* is defined as the below function.

$$y = 20.0 * |\sin x|$$

It is assumed that the angles for the sine function are measured in radians. In the program, both the values for *x* and *y* are computed based on input from the user. Once the values are found, the program should print them out as indicated by the examples. The program should also print out a graphical representation of the function, using a sequence of '\*'s to indicate the magnitude of *y*.

#### Lab Objectives

By the end of the lab, you should be able to:

- validate user input and handle invalid user input;
- create nested loops and loops with variable termination conditions;
- create and manipulate one-dimensional arrays.

#### Prerequisites

The lab deals with material from textbook chapter on loops and the sections pertaining to one-dimensional arrays.

#### What to Submit

The file **StarGraph.java** should be submitted to our course website for grading.

## Instructions

1. The class should be called **StarGraph**.
2. You must include a comment stating your name, the date, program purpose, and containing a statement of academic honesty as shown in Lab 02. When writing, you must also abide by the Java formatting and naming conventions.
3. The program should prompt the user for the following information:
  - the size  $N$  of the array (that is, the number values of  $x$  for which  $y$  will be computed);
  - $x_{min}$ , a minimum value for  $x$ ;
  - an *increment* for  $x$  (a value by which  $x$  is increased).

The messages to display to the user are shown in the examples below.

4. Note that the size of the arrays must be a positive integer. If the user enters 0 or a negative integer, then the program should print out an error message and terminate the program, as indicated in the examples.
5. The increment value should be a positive floating point value (a **double**). If the user enters 0 or a negative number, then the program should print an error message and terminate.
6. Once  $N$ ,  $x_{min}$ , and the increment are established, the program should create two double arrays, one to hold the  $x$  values and one to hold the  $y$  values. The size of both arrays will be  $N$ .

The first element of the  $x$ -array will hold  $x_{min}$ , and each successive element will be incremented by the value provided by the user. The values for the  $y$ -array will be defined using the corresponding value from the  $x$ -array and the function defined earlier. To compute the value for  $y$ , the program should use the **Math.abs** and **Math.sin** methods.

Observe that in order to populate the arrays with values, loops must be used.

7. Once computed, the values of  $x$  and  $y$  should be printed to the console. Each pair of values for  $x$  and  $y$  stored in the arrays should be printed. Note that in the examples, the values are formatted to have 3 digits after the decimal point.
8. After the values have been printed, a simple graph of the function should be plotted using asterisks (\*). Each successive line corresponds to an  $x$  value, and the number of '\*'s on the line corresponds to the whole-number part of  $y$  (the fractional part is ignored). To compute the number of '\*'s, you should not round the value for  $y$  (instead, truncate it).

## Submission and Grading

After you have completed and thoroughly tested **StarGraph.java**, submit it to our course webpage in order to receive credit for the lab (if you have any questions about how to submit this assignment, then ask your lab instructor for help at least 24 hours before this assignment is due). Always double check that your submission was successful on our course website.

The lab will be graded according to the following guidelines.

- A score between 0 and 100 will be assigned.



Please enter the number of x values to process: 10  
Enter a minimum value for x: 0  
Enter the amount to increment x: 0.31415926535

Please enter the number of x values to process: 10  
Enter a minimum value for x: 0  
Enter the amount to increment x: 0.31415926535

## Values

```
x: 0.000, y: 0.000
x: 0.314, y: 6.180
x: 0.628, y: 11.756
x: 0.942, y: 16.180
x: 1.257, y: 19.021
x: 1.571, y: 20.000
x: 1.885, y: 19.021
x: 2.199, y: 16.180
x: 2.513, y: 11.756
x: 2.827, y: 6.180
```

## Graph

[illegible]

```
Please enter the number of x values to process: 5
Enter a minimum value for x: 0
Enter the amount to increment x: 3.14159265359
```

```
Please enter the number of x values to process: 5
Enter a minimum value for x: 0
Enter the amount to increment x: 3.14159265359
```

## Values

```
x: 0.000, y: 0.000
x: 3.142, y: 0.000
x: 6.283, y: 0.000
x: 9.425, y: 0.000
x: 12.566, y: 0.000
```

### Graph

[illegible]

#### Example 4

Please enter the number of x values to process: 9

Enter a minimum value for x: -1.5707963267

Enter the amount to increment x: 0.78539816339

#### Values

x: -1.571, y: 20.000

x: -0.785, y: 14.142

x: 0.000, y: 0.000

x: 0.785, y: 14.142

x: 1.571, y: 20.000

x: 2.356, y: 14.142

x: 3.142, y: 0.000

x: 3.927, y: 14.142

x: 4.712, y: 20.000

#### Graph

```
.*****
.*****
.
.
.*****
.*****
.*****
.*****
.
.
.*****
.*****
```





```
Please enter the number of x values to process: 10
Enter a minimum value for x: -5
Enter the amount to increment x: 0.20
```

x: -5.000, y: 19.178  
x: -4.800, y: 19.923  
x: -4.600, y: 19.874  
x: -4.400, y: 19.032  
x: -4.200, y: 17.432  
x: -4.000, y: 15.136  
x: -3.800, y: 12.237  
x: -3.600, y: 8.850  
x: -3.400, y: 5.111  
x: -3.200, y: 1.167

```
. *****  
. *****  
.  
. *****  
.  
. *****  
.  
. *****  
.  
. *****  
.  
. *****  
.  
. *****
```





Please enter the number of x values to process: 10  
Enter a minimum value for x: 1.57079632679  
Enter the amount to increment x: 1.57079632679

### Values

```
x: 1.571, y: 20.000
x: 3.142, y: 0.000
x: 4.712, y: 20.000
x: 6.283, y: 0.000
x: 7.854, y: 20.000
x: 9.425, y: 0.000
x: 10.996, y: 20.000
x: 12.566, y: 0.000
x: 14.137, y: 20.000
x: 15.708, y: 0.000
```

## Graph

```

*****
.:
.:
*****
.:
.:
*****
.:
.:
*****
.:
.:
*****
.:
.:

```

```
Please enter the number of x values to process: 0
The number of x values must be an integer greater than 0.
```

```
Please enter the number of x values to process: 100
Enter a minimum value for x: -50
Enter the amount to increment x: 0
The increment must be a decimal number greater than 0.
```

```
Please enter the number of x values to process: 100
Enter a minimum value for x: -50
Enter the amount to increment x: 0
The increment must be a decimal number greater than 0.
```