Introduction to Array

Hui Chen

Department of Computer & Information Science

Brooklyn College

Objectives

- To describe why arrays are necessary in programming (§7.1)
- To declare array reference variables and create arrays (§§7.2.1–7.2.2).
- To obtain array size using **arrayRefVar.length** and know default values in an array (§7.2.3)
- To access array elements using indexes (§7.2.4)
- To declare, create, and initialize an array using an array initializer (§7.2.5).

Motivating Problem. Numbers above a statistic

1. Read 10 numbers, compute a statistic (e.g., the average), and find out how many numbers are above the statistic
2. Prompt the user to enter 10 numbers, compute a statistic (e.g., the average), and find out how many numbers are above the statistic

- Solutions?

Questions?

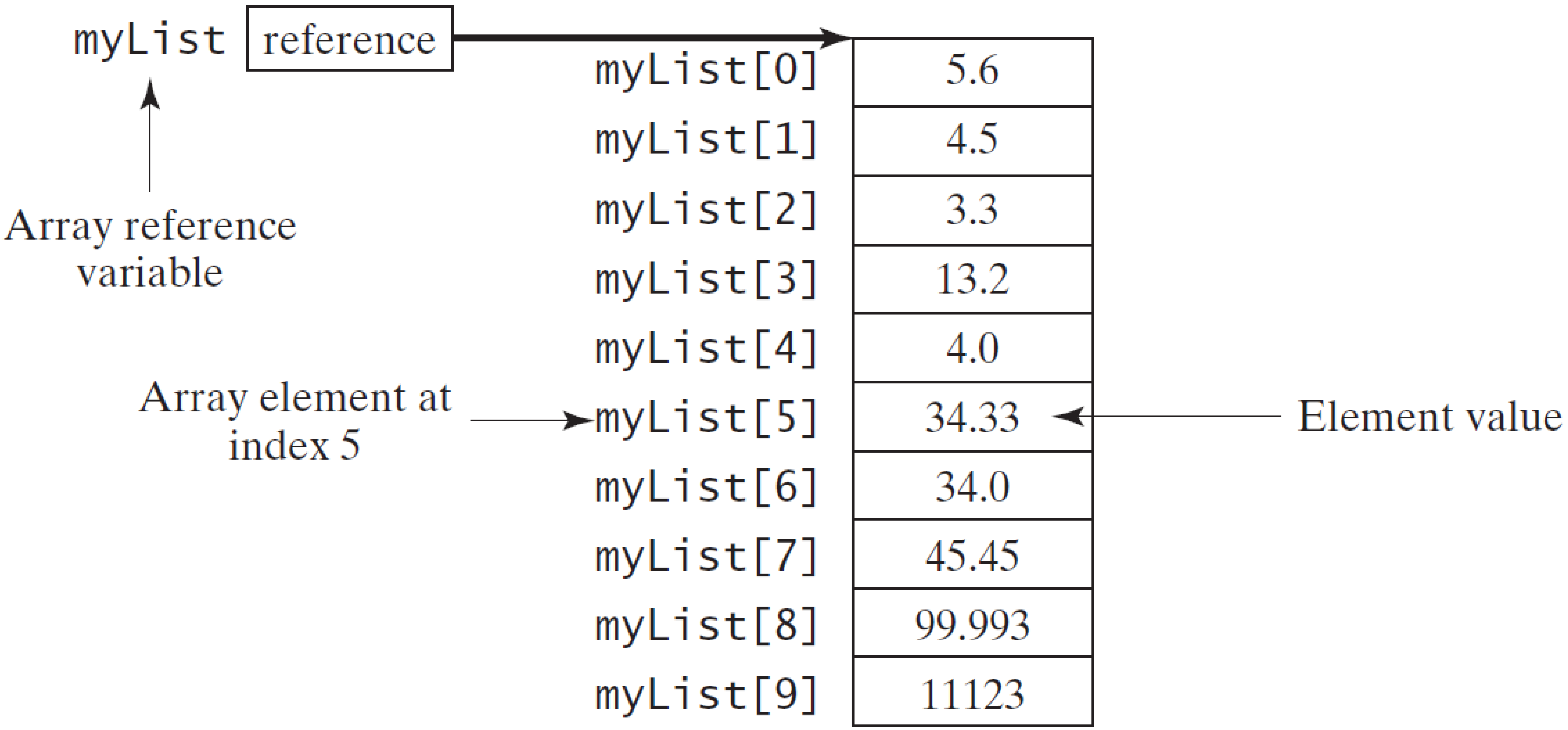
- Which solution do you like the best?
    - Ask the user to enter the numbers twice
    - Ask the user to enter the numbers once, and store and reuse the numbers (assume you can)

Concept of Array

- Array is a data structure that represents a collection of the same types of data.
    - Typically indexed and with equal random access time.

Example of Array

```
double[] myList = new double[10];
```



myList reference

Array reference variable

Array element at index 5

Element value

myList[0]	5.6
myList[1]	4.5
myList[2]	3.3
myList[3]	13.2
myList[4]	4.0
myList[5]	34.33
myList[6]	34.0
myList[7]	45.45
myList[8]	99.993
myList[9]	11123

Declaring Array Variables

- datatype[] arrayRefVar;
  - Example:

```
double[] myList;
```

- datatype arrayRefVar[]; // This style is allowed, but not preferred
  - Example:

```
double myList[];
```

Creating Arrays

- arrayRefVar = new datatype[arraySize];
- Example:
    - myList = new double[10];
- Creating an array means the memory space for the array is allocated, and does not mean it contains the desired values
    - What values does it contain? (to be discussed)

Referencing Array Elements

- Using index
  - myList[0] references the first element in the array.
  - myList[9] references the last element in the array.

Declaring and Creating in One Step

- datatype[] arrayRefVar = new datatype[arraySize];
- Example

```
double[] myList = new double[10];
```

- datatype arrayRefVar[] = new datatype[arraySize];
- Example

```
double myList[] = new double[10];
```

Questions?

- Declaring, creating, and referencing arrays

The Length of an Array

- Once an array is created, its size is fixed. It cannot be changed. You can find its size using

```
arrayRefVar.length
```

- Example,

```
myList.length returns 10
```

Default Values

- Creating an array means the memory space for the array is allocated, and does not mean it contains the desired values
- When an array is created, its elements are assigned the default value of
    - 0 for the numeric primitive data types,
    - '\u0000' for char types, and
    - false for boolean types
    - null for reference types (to be further explored in CISC3115)

Indexed Variables

- The array elements are accessed through the index.
- The array indices are *0-based*, i.e., it starts from 0 to arrayRefVar.length-1.
- Example
  - myList holds ten double values and the indices are from 0 to 9
- Each element in the array is represented using the following syntax, known as an *indexed variable*
  - arrayRefVar[index];

Using Indexed Variables

- After an array is created, an indexed variable can be used in the same way as a regular variable.
- Example

```
// adds the value in myList[0] and myList[1] to myList[2]
myList[2] = myList[0] + myList[1];
```

Questions?

Array Initializers

- Declaring, creating, initializing in one step:

```
double[] myList = {1.9, 2.9, 3.4, 3.5};
```

or

```
double[] myList = new double[] {1.9, 2.9, 3.4, 3.5};
```

- This shorthand syntax must be in one statement

Declaring, creating, initializing Using the Shorthand Notation

```
   double[] myList = {1.9, 2.9, 3.4, 3.5};
```

or

```
   double[] myList = new double[] {1.9, 2.9, 3.4, 3.5};
```

This shorthand notation is equivalent to the following statements:

```
   double[] myList = new double[4];
   myList[0] = 1.9;
   myList[1] = 2.9;
   myList[2] = 3.4;
   myList[3] = 3.5;
```

Errors and Pitfalls

- Using the shorthand notation, you have to declare, create, and initialize the array all in one statement. Splitting it would cause a syntax error.
- For example, the following is wrong:

```
double[] myList;
myList = {1.9, 2.9, 3.4, 3.5};
```

Questions

Question. What are the values?

```
public class Test {
  public static void main(String[] args) {
    int[] values = new int[5];
    for (int i = 1; i < 5; i++) {
      values[i] = i + values[i-1];
    }
    values[0] = values[1] + values[4];
    //  what are the values of the array elements?
  }
}
```

Questions?