# Java Programming AP Edition
## U3C7 1-D Array

ARRAY BASICS
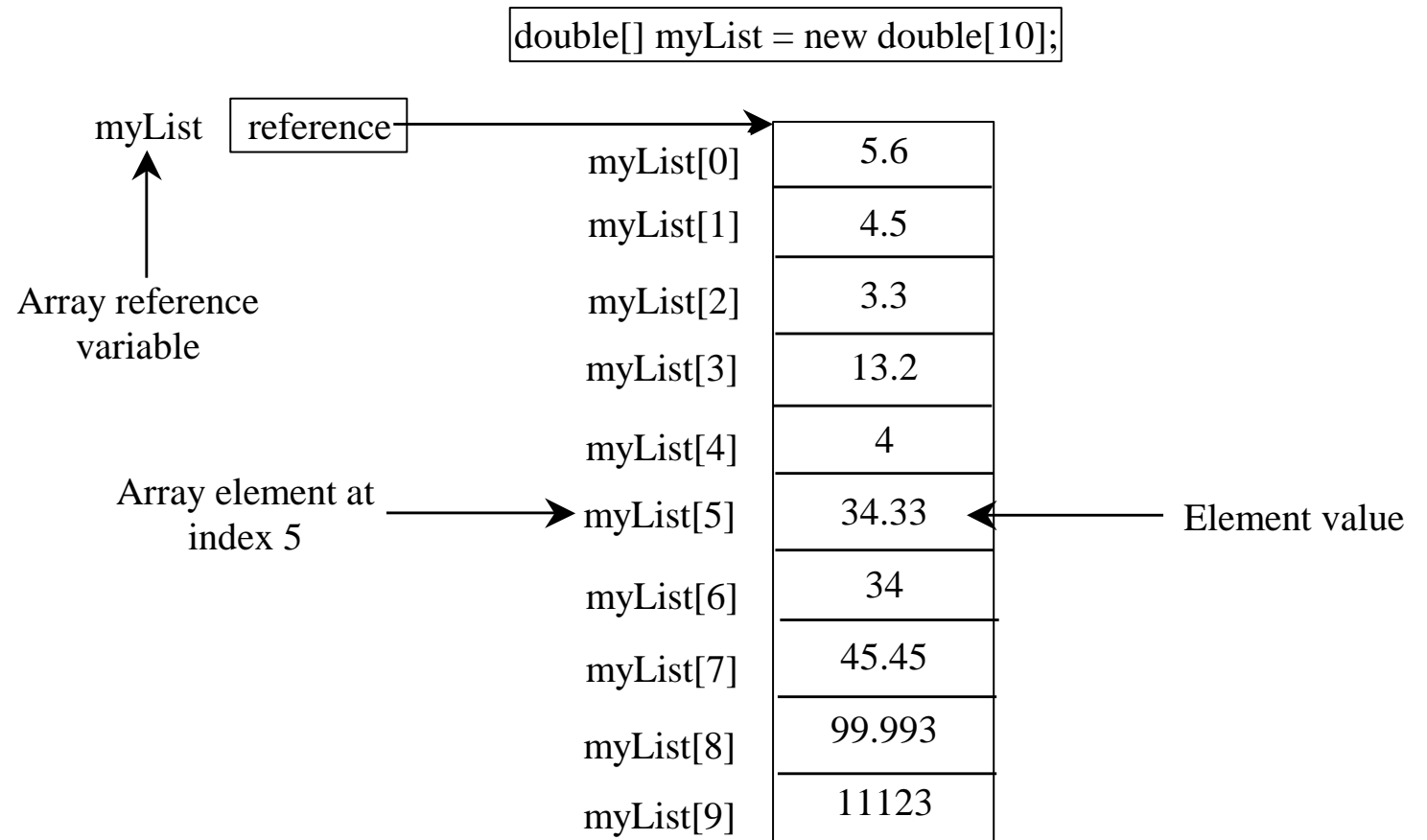
ERIC Y. CHOU, PH.D.        IEEE SENIOR MEMBER

# Introducing Arrays

Array is a data structure that represents a collection of the same types of data.

double[] myList = new double[10];

| | | |
|---|---|---|
| myList | reference | |

Array reference variable

Array element at index 5

| | |
|---|---|
| myList[0] | 5.6 |
| myList[1] | 4.5 |
| myList[2] | 3.3 |
| myList[3] | 13.2 |
| myList[4] | 4 |
| myList[5] | 34.33 |
| myList[6] | 34 |
| myList[7] | 45.45 |
| myList[8] | 99.993 |
| myList[9] | 11123 |

Element value

# Declaring Array Variables

- datatype[] arrayRefVar;

    Example:

    double[] myList;


- datatype arrayRefVar[]; <u>// This style is allowed, but not preferred</u>

    Example:

    double myList[];

# Creating Arrays

arrayRefVar = new datatype[arraySize];


Example:

myList = new double[10];


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];

    double[] myList = new double[10];

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

    double myList[] = new double[10];

# 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

For example,

myList.length returns 10

# Default 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.

# 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. In the example in Figure 6.1, 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. For example, the following code adds the value in myList[0] and myList[1] to myList[2].

```
myList[2] = myList[0] + myList[1];
```

# Array Initializers

- Declaring, creating, initializing in one step:

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

This shorthand syntax must be in one statement.

**myList = {2.9, 3.9, 4.4, 4.5}; // Compilation Error.**

# Declaring, creating, initializing Using the Shorthand Notation

double[] myList = {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;

# CAUTION

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};
```

# Trace Program with Arrays

Declare array variable values, create an array, and assign its reference to values

```java
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];
  }
}
```

After the array is created

| | |
|---|---|
| 0 | 0 |
| 1 | 0 |
| 2 | 0 |
| 3 | 0 |
| 4 | 0 |

# Trace Program with Arrays

i becomes 1

```java
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];
    }
}
```

After the array is created

| | |
|---|---|
| 0 | 0 |
| 1 | 0 |
| 2 | 0 |
| 3 | 0 |
| 4 | 0 |

# Trace Program with Arrays

i (=1) is less than 5

```java
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];
  }
}
```

After the array is created

| | |
|---|---|
| 0 | 0 |
| 1 | 0 |
| 2 | 0 |
| 3 | 0 |
| 4 | 0 |

# Trace Program with Arrays

After this line is executed, value[1] is 1

```
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];
    }
}
```

After the first iteration

| | |
|---|---|
| 0 | 0 |
| 1 | 1 |
| 2 | 0 |
| 3 | 0 |
| 4 | 0 |

# Trace Program with Arrays

After i++, i becomes 2

```java
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];
  }
}
```

After the first iteration

| | |
|---|---|
| 0 | 0 |
| 1 | 1 |
| 2 | 0 |
| 3 | 0 |
| 4 | 0 |

# Trace Program with Arrays

```java
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];
  }
}
```

i (= 2) is less than 5

After the first iteration

| | |
|---|---|
| 0 | 0 |
| 1 | 1 |
| 2 | 0 |
| 3 | 0 |
| 4 | 0 |

# Trace Program with Arrays

After this line is executed,
values[2] is 3 (2 + 1)

```java
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];
   }
}
```

After the second iteration

| | |
|---|---|
| 0 | 0 |
| 1 | 1 |
| 2 | 3 |
| 3 | 0 |
| 4 | 0 |

# Trace Program with Arrays

After this, i becomes 3.

```
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];
  }
}
```

After the second iteration

| | |
|---|---|
| 0 | 0 |
| 1 | 1 |
| 2 | 3 |
| 3 | 0 |
| 4 | 0 |

# Trace Program with Arrays

i (=3) is still less than 5.

```
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];
  }
}
```

After the second iteration

| | |
|---|---|
| 0 | 0 |
| 1 | 1 |
| 2 | 3 |
| 3 | 0 |
| 4 | 0 |

# Trace Program with Arrays

After this line, values[3] becomes 6 (3 + 3)

```
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];
  }
}
```

After the third iteration

| | |
|---|---|
| 0 | 0 |
| 1 | 1 |
| 2 | 3 |
| 3 | 6 |
| 4 | 0 |

# Trace Program with Arrays

After this, i becomes 4

```
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];
  }
}
```

After the third iteration

| | |
|---|---|
| 0 | 0 |
| 1 | 1 |
| 2 | 3 |
| 3 | 6 |
| 4 | 0 |

# Trace Program with Arrays

i (=4) is still less than 5

```
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];
  }
}
```

After the third iteration

| | |
|---|---|
| 0 | 0 |
| 1 | 1 |
| 2 | 3 |
| 3 | 6 |
| 4 | 0 |

# Trace Program with Arrays

After this, values[4] becomes 10 (4 + 6)

```java
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];
  }
}
```

After the fourth iteration

| | |
|---|---|
| 0 | 0 |
| 1 | 1 |
| 2 | 3 |
| 3 | 6 |
| 4 | 10 |

# Trace Program with Arrays

After i++, i becomes 5

```java
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];
  }
}
```

After the fourth iteration

| | |
|---|---|
| 0 | 0 |
| 1 | 1 |
| 2 | 3 |
| 3 | 6 |
| 4 | 10 |

# Trace Program with Arrays

i ( =5) < 5 is false. Exit the loop

```java
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];
  }
}
```

After the fourth iteration

| | |
|---|---|
| 0 | 0 |
| 1 | 1 |
| 2 | 3 |
| 3 | 6 |
| 4 | 10 |

# Trace Program with Arrays

After this line, values[0] is 11 (1 + 10)

```
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];
  }
}
```

| | |
|---|---|
| 0 | 11 |
| 1 | 1 |
| 2 | 3 |
| 3 | 6 |
| 4 | 10 |