COP-2210
Computer Programming I

Instructor: Dr. Antonio Hernandez

Text: Big Java: Early Objects, Interactive Edition, 6th Edition

Arrays

24. One Dimensional Arrays

Arrays: definition

Arrays:

- Groups together variables of the same data type.

- reference to every variable is done with a single name

- each member is accessed with an *index*

Arrays: definition

Arrays:

- format:

<data type> [ ] <name>;

<name> = new <data type>[<size>];

or

<data type> [ ] <name> = new <data type>[<size>];

Arrays: declaration

Example:

```
double [ ]  bills;
bills = new double [12]
```

or

```
double [ ] bills  = new double [12];
```

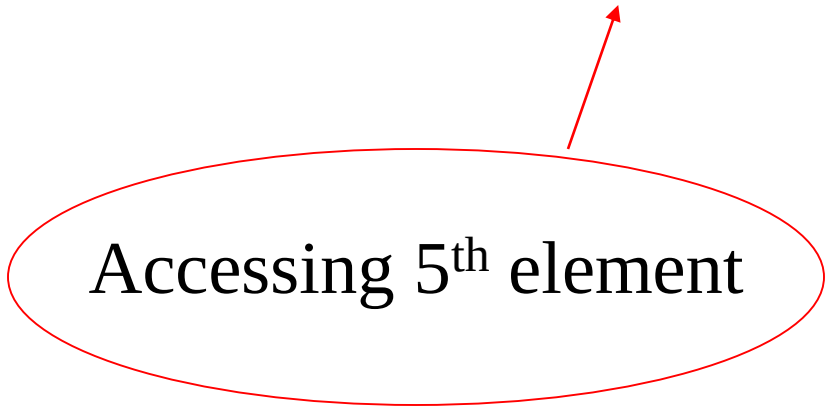
Arrays: access to elements

Arrays:

- Access: *<name> [<index>]*

Example:

… *bills[4]* …



Arrays: *Try it yourself*

```
//A simple array application

public class Prog24_01
{
   public static void main(String[] args)
   {
      double[] bills = new double[12];

      for (int i=0; i<12; i++)
      {
         bills[i] = 29.99;
      }

      for (int i=0; i<12; i++)
      {
         System.out.println(" Month " + (i+1) + ": $" + bills[i]);
      }
   }
}
```

Arrays: *Try it yourself*

```
public class Prog24_02
{
   public static void main(String[] args)
   {
      Scanner in = new Scanner(System.in);

      double[] bills = new double[12];

      for (int i=0; i<12; i++)
      {
         System.out.print("Enter the bill for month " + (i + 1) + ": $");
         bills[i] = in.nextDouble();
      }

      System.out.println("\nYour bills for the year are\n");

      for (int i=0; i<12; i++)
      {
         System.out.println(" Month " + (i+1) + ": $" + bills[i]);
      }
   }
}
```

Arrays: *Try it yourself*

```
import javax.swing.*;

public class Prog24_03
{
    public static void main (String args [ ])
    {
        double a[ ] = new double [5];

        // Fill the array with powers of two
        for ( int i = 0; i < 5;  i++)
        {
            a [i] = Math.pow(2, i);
        }

        // Prepare for displaying the values of a
        String s = "";

        for ( int i = 0; i < 5; i++)
        {
            s = s + ( int ) a[i] + "\n";
        }

        JOptionPane.showMessageDialog ( null, s, "Output",
                JOptionPane.INFORMATION_MESSAGE);
    }
}
```

Arrays: *Try it yourself*

```
import javax.swing.*;

public class Prog24_04
{
    public static void main (String args [ ])
    {
        double a[ ] = new double [5];

        // Fill the array with powers of two
        for ( int i = 0; i < a.length;  i++)
        {
            a [i] = Math.pow(2, i);
        }

        // Prepare for displaying the values of a
        String s = "";

        for ( int i = 0; i < a.length; i++)
        {
            s = s + ( int ) a[i] + "\n";
        }

        JOptionPane.showMessageDialog ( null, s, "Output",
                JOptionPane.INFORMATION_MESSAGE);
    }
}
```

***Size* of array a**

PRACTICE

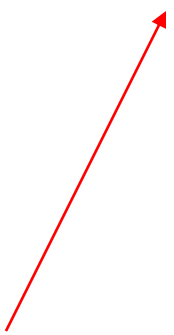
Program 24_05:

Write a Java program that declares an array of 10 integers and stores the number $\mathbf{n}^2$ in the location **n** of the array. Test the program by printing the list of values in the array.

Arrays: Initializing

Example:

```
String[ ] months  = { “Jan”, “Feb”, “Mar”,
                      “Apr”, “May”, “Jun”,
                      “Jul”,  “Aug”, “Sep”
                      “oct”,  “Nov”, “Dec”  };
```



No *new* or <*size*> is used

Initializing Arrays: *Try it yourself*

```
public class Prog24_06
{
    public static void main (String args [ ])
    {
            int[ ] xCoords = {10, -1, 15, 22, 34, -5, 17, 91, 0, 32};
            int[ ] yCoords = {6, 9, 86, -51, 20, 0, 7, 62, 30, 72};

             System.out.println("List of points: ");

            for ( int i = 0; i < xCoords.length;  i++)
            {
                System.out.println(" (" + xCoords[i] + ", " + yCoords[i] + ")");
            }
    }
}
```

PRACTICE

Program 24_07:

Write a Java program that declares an array a of 50 elements of type *double*. Populate the array with values this way:

- the first half of the elements are equal to the square of the index variable,
- the second half of the elements are equal to the square root of the index variable.

Arrays: *solution to PRACTICE 24_07*

```
public class Prog24_07 {
    public static void main(String[] args) {
        double[] a = new double[50];

        for (int i = 0; i < a.length/2; i++) {
            a[i] = Math.pow(i, 2);
        }

        for (int i = a.length/2; i< a.length; i++) {
            a[i] = Math.sqrt(i);
        }

        for (int i = 0; i < a.length; i++) {
            System.out.printf("%8.2f\n", a[i]);
        }

        System.out.println();
    }
}
```

PRACTICE

Program 24_08:

Modify the previous program so that in each line exactly 10 elements are displayed.

Arrays and Random Numbers: *Try it yourself*

```
//Prog24_09 Random numbers and arrays

import java.util.Random;

public class Prog24_09
{
   public static void main(String[] args)
   {
      Random rand = new Random();
      int[] randomIntegers = new int[10];

      for (int i = 0; i < randomIntegers.length; i++)
      {
         randomIntegers[i] = rand.nextInt(100);
         System.out.print(randomIntegers[i] + " ");
      }
      System.out.println();
   }
}
```

Random class

nextInt(a):
generates integer in [0, a)

PRACTICE

Program 24_10:

Write a Java application that fills an array with random integer numbers in [50, 90] and displays it. The program will calculate the average of the numbers in the array.