Al-Mustaqbal University College (MUC)

Intelligent Medical Systems Department

Computer Programing II

Lecturer: Maytham N. Meqdad

Lecture 1

ARRAY

An array is a special kind of object used to store a collection of data. An array differs from the other objects you have seen in two ways:

- All the data stored in an array must be of the same type. For example, you might use an array to store a list of values of type `double` that record rainfall readings in centimeters. Or you might use an array to store a list of objects of some class called `Species` that contain the records for various endangered species.
- An array object has only a small number of predefined methods. Because arrays were used by programmers for many years before classes were invented, they use a special notation of their own to invoke those few predefined methods, and most people do not even think of them as methods.

OBJECTIVES

After studying this chapter, you should be able to

- Describe the nature and purpose of an array
- Use arrays in simple Java programs
- Define methods that have an array as a parameter
- Define methods that return an array
- Use an array as an instance variable in a class
- Use an array that is not filled completely
- Order, or sort, the elements in an array
- Search an array for a particular item
- Define and use multidimensional arrays
- Insert text fields and text areas into your applets
- Draw arbitrary polygons in your applets

Suppose you want to compute the average temperature for the seven days in a week. You might use the following code:

```
Scanner keyboard = new Scanner(System.in);
System.out.println("Enter 7 temperatures:");
double sum = 0;
for (int count = 0; count < 7; count++)
{
   double next = keyboard.nextDouble();
   sum = sum + next;
}
double average = sum / 7;
```

Creating and Accessing Arrays

In Java, an array is a special kind of object, but it is often more useful to think of an array as a collection of variables of the same type. For example, an array consisting of a collection of seven variables of type double can be created as follows:

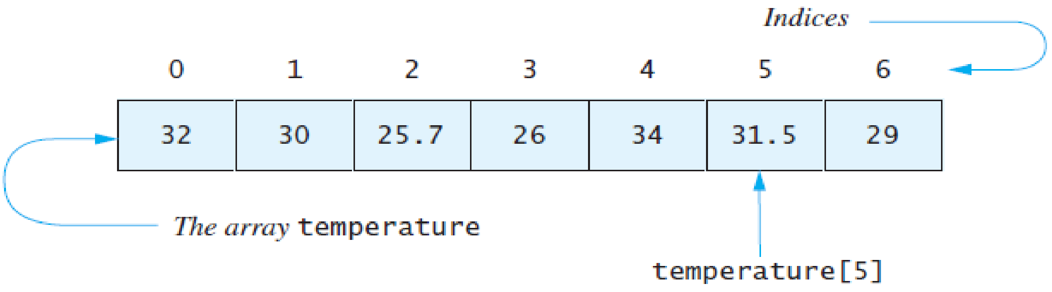
```
double[] temperature = new double[7];
```

This is like declaring the following seven strangely named variables to have the type double:

```
temperature[0], temperature[1], temperature[2], temperature[3],
temperature[4], temperature[5], temperature[6]
```

Variables like temperature[0] and temperature[1] that have an integer expression in square brackets are called **indexed variables**, **subscripted variable**, **array elements**, or simply **elements**. The integer expression within the square brackets is called an **index** or a **subscript**. Note that the numbering starts with 0, not 1.

An index is an integer expression that indicates an array element



Since an index can be an expression, we can write a loop to read values into the array temperature, as follows:

```
System.out.println("Enter 7 temperatures:");
for (int index = 0;index < 7; index++)
    temperature[index] = keyboard.nextDouble();
```

The user could type the seven values on separate lines or all on one line, separated by spaces. After the array values are read in, we can display them as follows:

```
System.out.println("The 7 temperatures are:");
for (int index = 0; index < 7; index++)
     System.out.print(temperature[index] + " ");
System.out.println( );
```

The program in Listing 7.1 shows an example that uses our sample array temperature as seven indexed variables, all of type double. Note that the program uses for loops similar to the ones we just considered.

```
/**
Reads 7 temperatures from the user and shows which are above
and which are below the average of the 7 temperatures.
*/
import java.util.Scanner;

public class ArrayOfTemperatures
{
    public static void main(String[] args)
    {
        double[] temperature = new double[7];

        // Read temperatures and compute their average:
        Scanner keyboard = new Scanner(System.in);
        System.out.println("Enter 7 temperatures:");
        double sum = 0;
        for (int index = 0; index < 7; index++)
        {
            temperature[index] = keyboard.nextDouble();
            sum = sum + temperature[index];
        }
        double average = sum / 7;
        System.out.println("The average temperature is " +
                           average);

        // Display each temperature and its relation to the average:
        System.out.println("The temperatures are");
        for (int index = 0; index < 7; index++)
        {
            if (temperature[index] < average)
                System.out.println(temperature[index] +
                                   " below average");
            else if (temperature[index] > average)
                System.out.println(temperature[index] +
                                   " above average");
            else //temperature[index] == average
                System.out.println(temperature[index] +
                                   " the average");
        }

        System.out.println("Have a nice week.");
    }
}
```

Sample Screen Output

```
Enter 7 temperatures:
32
30
25.7
26
34
31.5
29
The average temperature is 29.7428
The temperatures are
32.0 above average
30.0 above average
25.7 below average
26.0 below average
34.0 above average
31.5 above average
29.0 below average
Have a nice week.
```

Array Details

You create an array in the same way that you would create an object of a class type using the operation new, but the notation is slightly different. When creating an array of elements of type *Base_Type*, the syntax is as follows:

```
Base_Type[] Array_Name = new Base_Type[Length];
```

For example, the following creates an array named pressure that is equivalent to 100 variables of type int:

```
int[] pressure = new int[100];
```

Alternatively, the preceding can be broken down into two steps:

```
int[] pressure;
pressure = new int[100];
```

GOTCHA Assigning a Value to the Instance Variable length

Your program cannot assign a value to the instance variable length, as it is a final variable. For example, the following attempt to change the size of an array is invalid:

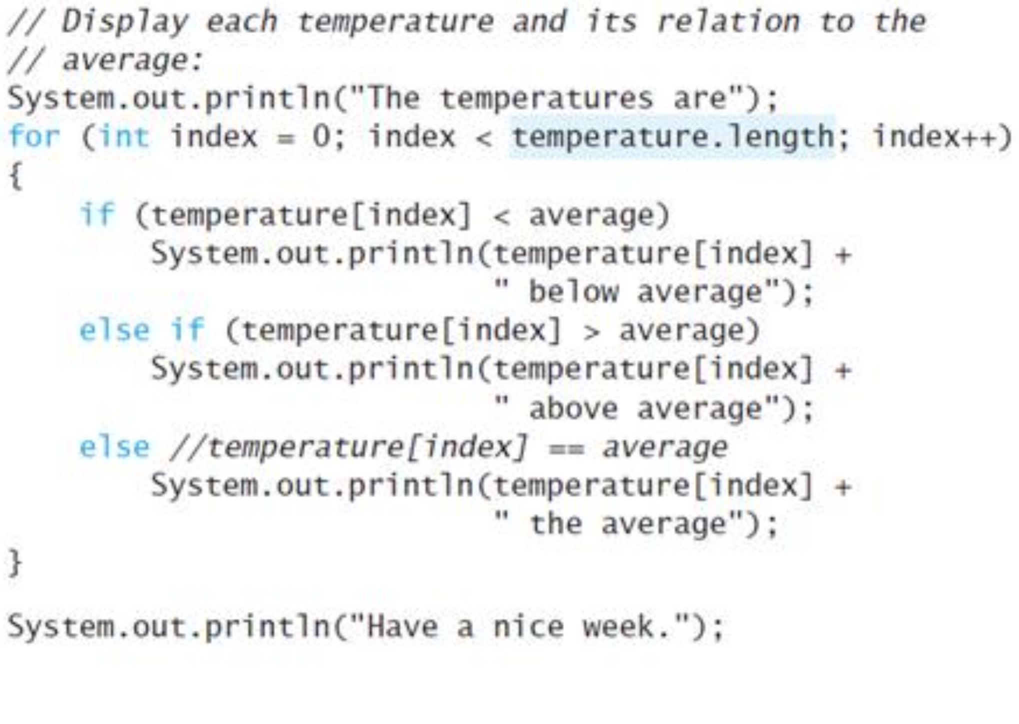
```
entry.length = 10; //Illegal!
```

In Listing 7.2 we have rewritten the program in Listing 7.1 using the instance variable length. We have also read the size of the array temperature from the user into the variable size. In this example, we could use size instead of temperature.length. However, since size is not final, its value can change and so might not always equal the value of temperature.length.

```java
/**
Reads temperatures from the user and shows which are above
and which are below the average of all the temperatures.
*/
import java.util.Scanner;

public class ArrayOfTemperatures2
{
    public static void main(String[] args)
    {
        Scanner keyboard = new Scanner(System.in);
        System.out.println("How many temperatures do you have?");
        int size = keyboard.nextInt( );
        double[] temperature = new double[size];

        // Read temperatures and compute their average:
        System.out.println("Enter " + temperature.length +
                           " temperatures:");
        double sum = 0;
        for (int index = 0; index < temperature.length; index++)
        {
            temperature[index] = keyboard.nextDouble();
            sum = sum + temperature[index];
        }
        double average = sum / temperature.length;
        System.out.println("The average temperature is " +
                            average);
```



```java
        // Display each temperature and its relation to the
        // average:
        System.out.println("The temperatures are");
        for (int index = 0; index < temperature.length; index++)
        {
            if (temperature[index] < average)
                System.out.println(temperature[index] +
                                   " below average");
            else if (temperature[index] > average)
                System.out.println(temperature[index] +
                                   " above average");
            else //temperature[index] == average
                System.out.println(temperature[index] +
                                   " the average");
        }

        System.out.println("Have a nice week.");
    }
}
```

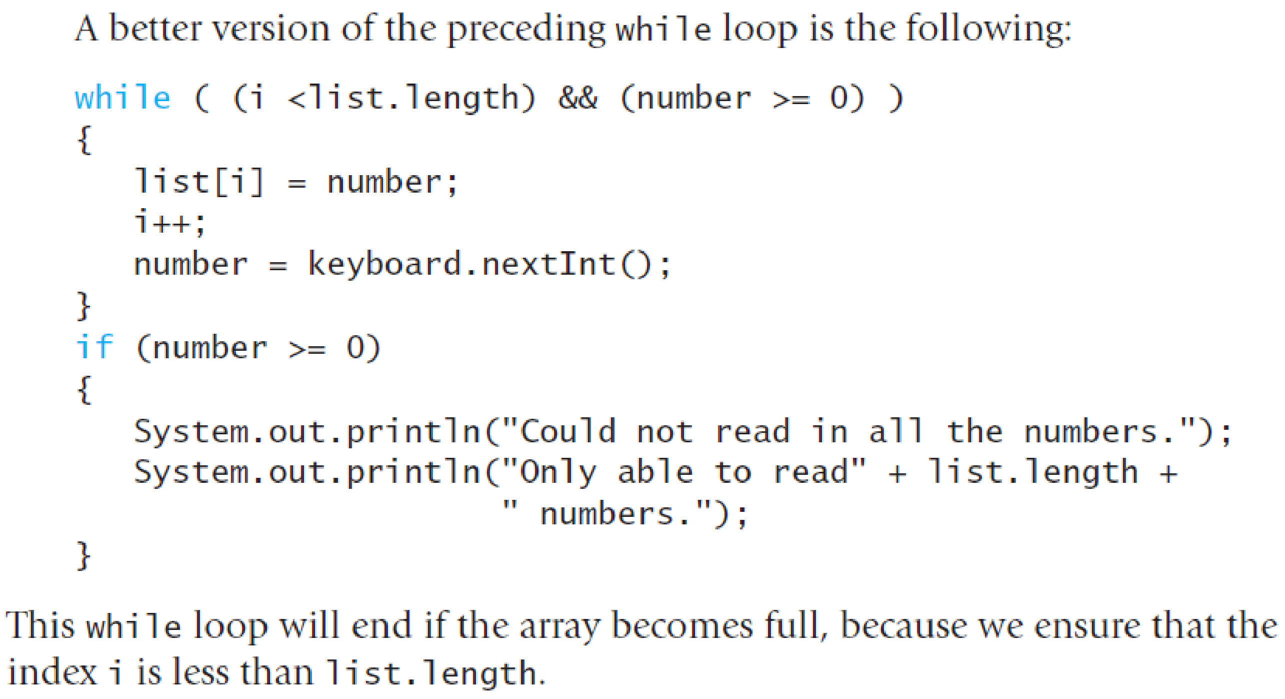
Sample Screen Output

```
How many temperatures do you have?
3
Enter 3 temperatures:
32
26.5
27
The average temperature is 28.5
The temperatures are
32.0 above average
26.5 below average
27.0 below average
Have a nice week.
```

One common way that array indices go out of bounds is when an array-processing loop is iterated one too many times. For example, let's consider a loop that fills an array. Suppose we want to read a sequence of nonnegative numbers from the keyboard, using a negative number as a sentinel value at the end of the data. We might use the following code:

```
System.out.println("Enter a list of nonnegative integers.");
System.out.println("Place a negative integer at the end.");
int[] list = new int[10];
Scanner keyboard = new Scanner(System.in);
int number = keyboard.nextInt();
int i = 0;
while (number >= 0)
{
   list[i] = number;
   i++;
   number = keyboard.nextInt();
}
```

If the user enters more numbers than can fit in the array, this code produces an array index that is out of bounds.



A better version of the preceding `while` loop is the following:

```
while ( (i <list.length) && (number >= 0) )
{
   list[i] = number;
   i++;
   number = keyboard.nextInt();
}
if (number >= 0)
{
   System.out.println("Could not read in all the numbers.");
   System.out.println("Only able to read" + list.length +
                      " numbers.");
}
```

This `while` loop will end if the array becomes full, because we ensure that the index `i` is less than `list.length`.

- To calculate the average weekly glucose level for a diabetic patient, we can write the following code in Java by using array:-

```
import java.util.Scanner;

public class DiabetesPatient {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        double[] glucoseLevels = new double[7];

        double sum = 0;


        for (int i = 0; i < glucoseLevels.length; i++) {

            System.out.print("Enter glucose level for day " +
(i+1) + ": ");
```

```
            glucoseLevels[i] = input.nextDouble();
            sum += glucoseLevels[i];
        }

        double weeklyAverage = sum / glucoseLevels.length;
        System.out.println("Weekly average glucose level is: "
+ weeklyAverage);
    }
}
```

- **To calculate the average high and low blood pressure for an unlimited number using the array in the Java language, we write the following code:**

```
import java.util.Scanner;
public class BloodPressure {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("Enter the number of blood pressure readings: ");
        int numReadings = input.nextInt();
        int[] highPressures = new int[numReadings];
        int[] lowPressures = new int[numReadings];
        int highSum = 0;
        int lowSum = 0;

        for (int i = 0; i < numReadings; i++) {
            System.out.print("Enter high pressure reading " + (i+1) + ": ");
            highPressures[i] = input.nextInt();
            System.out.print("Enter low pressure reading " + (i+1) + ": ");
            lowPressures[i] = input.nextInt();
            highSum += highPressures[i];
            lowSum += lowPressures[i];
        }

        double highAverage = (double) highSum / numReadings;
        double lowAverage = (double) lowSum / numReadings;
        System.out.println("Average high pressure reading: " + highAverage);
        System.out.println("Average low pressure reading: " + lowAverage);
    }
}
```

-Write program in Java using array to print the names of each students, calculate the total score for each student, and calculate the GPA for each student.

```
import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        // استخدم المتغيرات التالية لتخزين بيانات الطلاب
        int numOfStudents, totalMarks;
        String studentName;

        // اطلب من المستخدم إدخال عدد الطلاب
```

```
System.out.print("Enter the number of students: ");
numOfStudents = scanner.nextInt();

// قم بإنشاء مصفوفة لتخزين بيانات الطلاب
int[] studentMarks = new int[numOfStudents];
String[] studentNames = new String[numOfStudents];
double[] studentAverages = new double[numOfStudents];

// اطلب من المستخدم إدخال بيانات الطلاب وحساب المجموع والمعدل
for (int i = 0; i < numOfStudents; i++) {
    System.out.print("Enter the name of student " + (i+1) + ": ");
    studentName = scanner.next();
    studentNames[i] = studentName;

    totalMarks = 0;
    for (int j = 0; j < 3; j++) { // نفترض أن كل طالب لديه ٣ درجات
        System.out.print("Enter the mark for test " + (j+1) + ": ");
        int mark = scanner.nextInt();
        totalMarks += mark;
    }
    studentMarks[i] = totalMarks;

    studentAverages[i] = totalMarks / 3.0;
}

// اطبع بيانات الطلاب
for (int i = 0; i < numOfStudents; i++) {
    System.out.println("Student " + (i+1) + ": " + studentNames[i]);
```

```
            System.out.println("Total marks: " + studentMarks[i]);

            System.out.println("Average: " + studentAverages[i]);

            System.out.println();

        }

    }

}
```