Arrays
(Savitch, Chapter 7)

TOPICS

- Array Basics
- Arrays in Classes and Methods
- Programming with Arrays
- Searching and Sorting Arrays
- Multi-Dimensional Arrays
- Static Variables and Constants

Creating and Accessing Arrays

- An array is a special kind of object
- Think of as collection of variables of same type
- Creating an array with 7 variables of type double
  
  ```java
  double[] temperature = new double[7];
  ```
- To access an element use
  - The name of the array
  - An index number enclosed in braces
- Array indices begin at zero

Creating and Accessing Arrays

- Figure 7.1  A common way to visualize an array

- Note `class ArrayOfTemperatures`
Creating and Accessing Arrays

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

• Syntax for declaring an array with `new`

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

• The number of elements in an array is its length
• The type of the array elements is the array's base type

Square Brackets with Arrays

• With a data type when declaring an array:
  `int [] pressure;` <- Java convention
  `int pressure [];` <- C/C++ convention
• To enclose an integer expression to declare the length of the array:
  `pressure = new int[100];`
• To access the array with an index value:
  `pressure[3] = keyboard.nextInt();`
The Instance Variable \textit{length}

• As an object an array has only one public instance variable:
  – Variable \textit{length}
  – Contains number of elements in the array
  – It is final, value cannot be changed

More About Array Indices

• Index of first array element is 0
• Last valid Index is \textit{arrayName.length} - 1
• Array indices must be within bounds to be valid:
  – When program tries to access outside bounds, run time exception occurs
• Get used to using index 0

Initializing Arrays

• Possible to initialize at declaration time
  \texttt{double[]} \texttt{reading} = \{3.3, 15.8, 9.7\};

• Also may use normal assignment statements
  – One at a time
  – In a loop
    \texttt{int[]} \texttt{count} = \texttt{new int[100]};
    \texttt{for} (\texttt{int} \texttt{i} = 0; \texttt{i} < 100; \texttt{i}++)
    \texttt{count[i]} = 0;
Indexed Variables as Method Arguments

• Indexed variable of an array
  – Example \( a[i] \)
  – Can be used anywhere variable of array base type can be used
  – Example

Entire Arrays as Arguments

• Declaration of array parameter similar to how an array is declared
• Example:

```java
public class SampleClass {
    public static void incrementArrayBy2(double[] anArray) {
        for (int i = 0; i < anArray.length; i++) {
            anArray[i] = anArray[i] + 2;
        }
    }
}
```

• Array parameter in a method heading does not specify the length
  – An array of any length can be passed to the method
  – Inside the method, elements of the array can be changed
• When you pass the entire array, do not use square brackets in the argument.
  – Kind of like you only pass the variable name of a String or an integer, as well.
Arguments for Method main

- Recall heading of method `main`
  ```java
  public static void main (String[] args)
  ```
- This declares an array
  - Formal parameter named `args`
  - Its base type is `String`
- Thus possible to pass to the run of a program multiple strings
  - These can then be used by the program

Array Assignment and Equality

- Arrays are objects
  - Assignment and equality operators behave (misbehave) as specified in previous chapter
- Variable for the array object contains memory address of the object
  - Assignment operator `=` copies this address
  - Equality operator `==` tests whether two arrays are stored in same place in memory
- Example

Array Assignment and Equality

- Note results of `==`
- Note definition and use of the static method `equals` from the Arrays class
  - Receives two array parameters
  - Checks length and each individual pair of array elements
- Remember array types are reference types

Methods that Return Arrays

- A Java method may return an array
- Note definition of return type as an array
- To return the array value
  - Declare a local array
  - Use that identifier in the `return` statement
- Example
Programming Example: A Specialized List Class

- Objects can be used for keeping lists of items
- Methods include
  - Capability to add items to the list
  - Also delete entire list, start with blank list
  - But no method to modify or delete list item
- Maximum number of items can be specified

Partially Filled Arrays

- Array size specified at definition
- Not all elements of the array might receive values
  - This is termed a partially filled array
- Programmer must keep track of how much of array is used