Contents

1. Introduction
2. Types and Variables
3. Statements and Control Flow
4. Reading Input
5. Classes and Objects
6. Arrays
7. Methods
8. Scope and Lifetime
9. Utility classes
10. Introduction to Object-Oriented Analysis and Design
Chapter 6: Arrays

After this chapter you will be able to:

- Declare and initialize array variables
- Create array objects
- Use command-line arguments
Declaring Array Variables

• Array variables are declared using the square brackets ([]) 

```java
int [] a;    // array of ints
String [] b; // array of Strings
```

• Arrays are objects, and therefore array variables store **addresses**
Creating Array Objects

• Use `new` to create an array object

```java
int[] a = new int[3];
```

```
a   [ ]   [ ]   [ ]
```

• Indexing starts at 0

```java
a[0] = 5;
a[1] = 6;
a[2] = 7;
```

```
a   5 6 7
```
Array Initialization

- Can use braces {} for array initialization (new is implicitly called)

```java
int [] a = {5, 6, 7};
String [] b = {"hi", "there"};
```

Diagram:
- `a` arrow pointing to `5 6 7`
- `b` arrow pointing to `hi there`
Array Length

• Arrays know their own length

```java
int [] a = {5,6,7};

for (int i = 0; i < a.length; i++)
    System.out.println("a[" + i + "] = " + a[i]);
```

Output

```
a[0] = 5
a[1] = 6
a[2] = 7
```
Array Assignment

• Assignment between array variables copies addresses

```c
int [] a = {5, 6, 7};
int [] b = {8, 9, 2};

a = b;
```

• Question: what will happen to the memory containing 5, 6, 7?
Array Copying

• `System.arraycopy` copies array elements

```java
int [] a = {5,6,7};
int [] b = {8,9,2};

System.arraycopy(b,0,a,0,b.length);
```
Command Line Arguments

- Command line arguments are passed to `main` as an array of Strings

```java
public class Args {
    public static void main (String [] args) {
        for (int i =0; i < args.length; i++)
            System.out.println(args[i]);
    }
}
```

Command `how are you today`:
- `args[0]`
- `args[1]`
- `args[2]`
- `args[3]`
Converting Strings to Integers

- `Integer.parseInt` converts a string value to an integer value

```java
public class Sum {
    public static void main (String [] args) {
        int sum = 0;
        for (int i =0; i < args.length; i++)
            sum += Integer.parseInt(args[i]);
        System.out.println("total sum is: " + sum)
    }
}
```

Command: Sum 3 4 5 6

Output: total sum is: 18
Two Dimensional Arrays

• Two dimensional array variables are declared with two pairs of brackets ([][])

```
int [][] a = new int[3][3];
for (int i = 0; i < 3; i++)
    for (int j = 0; j < 3; j++)
        a[i][j] = 1;
```
Two Dimensional Array Init

• Nesting braces ({{}}) can be used for initialization

```java
int [] [] a = {{1,1,1}, {1,2}, {3,4,5}};
```

![Diagram of two-dimensional array initialization]
public class Search {
    public static void main (String [] args) {
        String [] table = {"apple", "orange","pear"};
        for (int i =0; i < table.length; i++)
            if (args[0].equals(table[i])) {
                System.out.println("found "+ args[0]);
                return;
            }
        System.out.println(args[0] + "not found");
    }
}
Traversing a 2D Array

```
public class Search {
    public static void main (String [] args) {
        String [][] words = {{"apple", "orange", "pear"},
                            {"banana", "mango"}};
        for (int i =0; i < words.length; i++)
            for (int j =0; j< words[i].length; j++)
                System.out.println(words[i][j]);
    }
}
```

- What is the output?
public class Sort {
    public static void main (String [] args) {
        int [] elements = new int[args.length];

        for (int i = 0; i < args.length; ++i)
            elements[i] = Integer.parseInt(args[i]);

        for (int i = 0; i < elements.length-1; i++)
            for (int j = i + 1; j > 0; j--)
                if (elements[j] < elements[j-1]) { // swap
                    int temp = elements[j-1];
                    elements[j-1] = elements[j];
                    elements[j] = temp;
                }

        for (int i = 0; i < elements.length; i++)
            System.out.print(elements[i] + "\t");

        System.out.println();
    }
}
Sorting Command Line Args

Command: Sort 6 5 3 7

Output: 3 5 6 7
It’s Exercise Time