

Homework Eight

A201/A597/I210—Spring Semester 2005

Due in OnCourse by Tuesday, April 5, 11:59pm
(*No written report is due in class, but read below for details!*)

Abstract

Read and solve the problems below. Turn in a document in OnCourse. The Computer Science Department¹ and the School of Informatics² clearly specify the rules of academic honesty and academic integrity, so please read the documents and make sure you comply.

Posting solutions or major hints on the bulletin board is not allowed.

1 The Summary

1. An array is a structured data type with a fixed number of components. Every component is of the same type, and the components are accessed using their relative positions in the array.
2. Elements of a one-dimensional array are arranged in the form of a list. A two-dimensional array is an array in which the elements are arranged in table form (with rows and columns).
3. To access the elements of a two-dimensional array you need to use an index. An array index can be any expression that evaluates to a non-negative integer. The value of the index must always be less than the size of the array.
4. In Java, an array index starts with 0
5. In Java, [] is an operator, called the array subscripting operator.
6. When an array object is instantiated, its components are initialized to their default values.
7. Arrays can be initialized when they are created.
8. Associated with each array that has been instantiated (that is, memory has been allocated to store the data), there is a `public` instance variable `length`. The variable `length` contains the size of the array.
9. If an array goes out of bounds (index used is \geq array's length) the program throws an `ArrayIndexOutOfBoundsException`.
10. Like any object arrays can be passed as parameters to methods.

¹<http://www.cs.indiana.edu/Academics/integrity.html>

²<http://www.informatics.indiana.edu/courses/honesty.asp>

11. Individual array components can be passed as parameters to methods (which, of course, also comes as no surprise).
12. You can create an array of objects.
13. Two-dimensional arrays are arrays of arrays. Each row, that is, is a one-dimensional array.
14. When an array is instantiated, the elements are given initial values automatically, depending on the data type. Numeric types are set to 0; `boolean` elements to `false`; `char` elements to `32` (a space), and object references (reference types, user-defined types) are set to `null`.

2 What to Turn In.

Like in Homework Six and Seven you will not have to turn in a written report in lecture any longer. Instead you will have to describe your thinking in a document that you need to turn in to OnCourse. Just answer the questions below. Be sure to explain your answers.

(The document could be in PDF, Word format or plain text).

3 The Problems

3.1 Reading and Understanding Code

1. What is the output of this code sequence?

```
double[] a = {12.5, 48.3, 65.0};
System.out.println(a[1]);
```

2. What is the output of this code sequence?

```
int[] a = new int[6];
System.out.println(a[4]);
```

3. What is the output of this code sequence?

```
double[] a = {12.5, 48.3, 65.0 };
System.out.println(a.length);
```

4. What is the output of this code sequence?

```
int[] a = {12, 48, 65};
for (int i = 0; i < a.length; i++)
    System.out.println(a[i]);
```

5. What is the output of this code sequence?

```
int[] a = { 12, 48, 65 };
for (int i = 0; i < a.length; i++)
    System.out.println("a[" + i + "] = " + a[i]);
```

6. What is the output of this code sequence?

```
int s = 0;
int[] a = {12, 48, 65};
for (int i = 0; i < a.length; i++)
    s += a[i];
System.out.println("s = " + s);
```

7. What is the output of this code sequence?

```
int[] a = new int[10];
for (int i = 0; i < a.length; i++)
    a[i] = i + 10;
System.out.println(a[4]);
```

8. What is the output of this code sequence?

```
double[] a = {12.3, 99.6, 48.2, 65.8};
double temp = a[0];
for (int i = 0; i < a.length; i++) {
    if (a[i] > temp)
        temp = a[i];
}
System.out.println(temp);
```

9. What is the output of this code sequence?

```
int[] a = { 12, 48, 65, 23 };
int temp = a[1];
a[1] = a[3];
a[3] = temp;
for (int i = 0; i < a.length; i++)
    System.out.print(a[i] + " ");
```

10. What does this method do?

```
public int foo(int[] a) {
    int temp = 0;
    for (int i = 0; i < a.length; i++) {
        if (a[i] == 5)
            temp++;
    }
    return temp;
}
```

11. What does this method do?

```
public int foo(int[] a) {
    for (int i = 0; i < a.length; i++) {
        if (a[i] == 10)
            return i;
    }
    return -1;
}
```

12. What does this method do?

```
public boolean foo(int[] a) {
    for (int i = 0; i < a.length; i++) {
        if (a[i] < 0)
            return false;
    }
    return true;
}
```

13. What does this method do?

```
public String[] foo(String[] a) {
    String[] temp = new String[a.length];
    for (int i = 0; i < a.length; i++) {
        temp[i] = a[i].toLowerCase();
    }
    return temp;
}
```

14. What does this method do?

```
public boolean[] foo(String[] a) {
    boolean[] temp = new boolean[a.length];
    for (int i = 0; i < a.length; i++) {
        if (a[i].contains("@"))
            temp[i] = true;
        else
            temp[i] = false;
    }
    return temp;
}
```

15. Consider the following declaration: `int[][] beta = new int[3][3];`
What's stored in `beta` after each of the following statements execute?

```
for (i = 0; i < 3; i++)
    for (j = 0; j < 3; j++)
        beta[i][j] = 0;
```

```
for (i = 0; i < 3; i++)
    for (j = 0; j < 3; j++)
        beta[i][j] = i + j;
```

```
for (i = 0; i < 3; i++)
    for (j = 0; j < 3; j++)
        beta[i][j] = i * j;
```

```
for (i = 0; i < 3; i++)
    for (j = 0; j < 3; j++)
        beta[i][j] = 2 * (i + j) % 4;
```

3.2 Fill in the Code

For all the exercises in this section fill in the missing code to make the snippet do what the description says it should do.

1. This code assigns the value of 10 to all the elements of an array **a**.

```
int[] a = new int[25];
for (int i = 0; i < a.length; i++) {
    // your code goes here
}
```

2. This code prints all the elements of array **a** that have a value greater than 20.

```
double[] a = { 45.2, 13.1, 12.8, 87.4, 99.0, 100.1, 43.8, 2.4 };
for (int i = 0; i < a.length; i++) {
    // your code goes here
}
```

3. This code prints the average of the elements of array **a**.

```
int[] a = {45, 13, 12, 87, 99, 100, 43, 2};
double average = 0.0;
for (int i = 0; i < a.length; i++) {
    // your code goes here
}
// ... and your code continues here
```

4. This code calculates and prints the dot product of two arrays **a** and **b** using the formula $\sum_{i=0}^{a.length} (a[i] * b[i])$ (notice the boundary condition implies the two arrays must be of same length).

```
int[] a = {3, 7, 9};
int[] b = {2, 9, 4};
int dotProduct = 0;
for (int i = 0; i < a.length; i++) {
    // your code goes here
}
```

5. This code prints the following three lines

```
a[0] = 3
a[1] = 6
a[2] = 10

int[] a = { 3, 6, 10 };

for (int i = 0; i < a.length; i++) {
    // your code goes here
}
```

6. This method returns `true` if an element in an array of `Strings` passed as a parameter contains the substring `IBM`; otherwise it returns `false`.

```
public boolean foo(String[] a) {  
    // your code goes here  
}
```

7. This method returns the number of elements in an array passed as a parameter that are multiples of 7

```
public int foo(int[] a) {  
    // your code goes here  
}
```

8. This method returns `true` if the first two elements of the array passed as a parameter have the same value; otherwise, it returns `false`.

```
public boolean foo(String[] args) {  
    // your code goes here  
}
```

3.3 Identifying Errors in Code

1. Where is the error in this code sequence?

```
double[] a = { 3.3, 26.0, 48.4 };  
a[4] = 2.5;
```

2. Where is the error in this code sequence?

```
double[] a = { 3.3, 26.0, 48.4 };  
System.out.println(a[-1]);
```

3. Where is the error in this code sequence?

```
double[] a = { 3.3, 26.0, 48.4 };  
System.out.print(a{1});
```

4. Where is the error in this code sequence?

```
double[] a = { 3.3, 26.0, 48.4 };  
for (int i = 0; i <= a.length; i++)  
    System.out.println(a[i]);
```

5. Where is the error in this code sequence?

```
double a[3] = { 3.3, 26.0, 48.4 };
```

6. Where is the error in this code sequence?

```
int a[] = { 3, 26, 48, 5 };  
int b[] = { 3, 26, 48, 5 };  
if (a != b)  
    System.out.println("Array elements are NOT identical.");
```

7. Where is the error in this code sequence?

```
int[] a = { 3, 26, 48, 5 };  
a.length = 10;
```

8. Where is the logic error in this code sequence?

```
int[] a = { 3, 26, 48, 5 };  
System.out.println("The array elements are " + a);
```