Lab Assignment Three

C212/A592—Fall Semester 2010

Due in OnCourse by Friday, September 17, 11:55pm
(Dropbox will stay open until Saturday, September 18, 11:55pm)

Abstract
Read and solve the problems below. Turn in a document to OnCourse.

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 one-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. To access the elements of a two-dimensional array you need to use two indices, one for the row and the other one for the column.

5. In Java, an array index starts with 0

6. In Java, [ ] is an operator, called the array subscripting operator.

7. When an array object is instantiated, its components are initialized to their default values.

8. Arrays can be initialized to some other custom value than the default when they are created.

9. Associated with each array that has been instantiated (by instantiated we mean that memory has been allocated to store the data), there is a public instance variable length. The variable length contains the size of the array.

10. If an array goes out of bounds (index used is ≥ array’s length) the program throws an ArrayIndexOutOfBoundsException.

11. Like any object arrays can be passed as parameters to methods. This simply wants to emphasize that we have seen that already: main’s argument is an array of Strings traditionally named args. With this the mystery about main is starting to diminish.
12. Individual array components can be passed as parameters to methods (which, of course, also comes as no surprise).
13. You can create an array of objects (as opposed to, say, an array of \texttt{int}s or values of other primitive type.)
14. Two-dimensional arrays are arrays of arrays: each row in a two-dimensional array is a one-dimensional array.
15. When an array is instantiated, the elements are given initial values automatically, depending on the data type. Numeric types are set to 0; \texttt{boolean} elements to \texttt{false}; \texttt{char} elements to 32 (a space), and object references (values of reference type) are set to \texttt{null}.

2 What to Turn In.

Like in the first two lab assignments you need to describe your thinking in a document that must be turned into OnCourse. Answer the questions below. Be sure to explain your answers. The document you upload could be in PDF, Word format or plain text.

Peer tutoring starts this Thursday in LH112 (with snacks and everything!). Lab is due Fri at midnight but the box will stay open one extra day in case you can’t make the deadline. If you experience any problems or unexpected delays be sure to contact me\textsuperscript{2} and let me know.

3 The Problems

3.1 Reading and Understanding Code

1. What is the output of this code sequence?

\begin{verbatim}
    double[] a = {12.5, 48.3, 65.0};
    System.out.println(a[1]);
\end{verbatim}

2. What is the output of this code sequence?

\begin{verbatim}
    int[] a = new int[6];
    System.out.println(a[4]);
\end{verbatim}

3. What is the output of this code sequence?

\begin{verbatim}
    double[] a = {12.5, 48.3, 65.0};
    System.out.println(a.length);
\end{verbatim}

4. What is the output of this code sequence?

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

\textsuperscript{1}Integers are values of primitive types, objects are values of reference type. We will start talking about objects next week. Strings are objects and the argument to \texttt{main} is an array of objects of type \texttt{String}. The objects in that array are of type \texttt{String}; they are collected from the command line when you run the program.

\textsuperscript{2}dgerman@indiana.edu
5. What is the output of this code sequence?

```java
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?

```java
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?

```java
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?

```java
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?

```java
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?

```java
public static void main(String[] args) {
    int temp = 0;
    for (int i = 0; i < args.length; i++)
        if (Integer.parseInt(args[i]) == 5)
            temp++;
    System.out.println(temp);
}
```

11. What does this method do?

```java
public static int fun(int[] args) {
    for (int i = 0; i < args.length; i++)
        if (args[i] == 10)
            return i;
    return -1;
}
```
12. What does this method do?

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

13. What does this method do?

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

Try it like this:

```java
class One {
    public static void main(String[] args) {
        nuf(args);
    }
}
```

14. What do you think this method does?

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

Try it like this:

```java
class One {
    public static void main(String[] args) {
        nuf(fun(args));
    }
    public static String[] fun(String[] a) {
        String[] temp = new String[a.length];
        for (int i = 0; i < a.length; i++)
            temp[i] = a[i].toLowerCase();
        return temp;
    }
    public static void nuf(String[] a) {
        for (int i = 0; i < a.length; i++)
            System.out.println(a[i])
    }
}
```
15. What does this method do? How can you test it?

```java
public static boolean[] fun(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;
}
```

The String documentation can be found here³.

16. Consider the following declaration: `int[][] beta = new int[3][3];`

What’s stored in `beta` after each of the following statements execute?

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

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

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

```java
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`.

   ```java
   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.

   ```java
   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
   }
   ```

³http://download.oracle.com/javase/6/docs/api/java/lang/String.html
3. This code prints the average of the elements of array `a`.

   ```java
   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 $\sum_{i=0}^{\text{length}} (a[i] \cdot b[i])$ (notice the boundary condition implies the two arrays must be of same length).

   ```java
   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

   ```java
   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 `String` passed as a parameter contains the substring `IBM`; otherwise it returns `false`.

   ```java
   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.

   ```java
   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`.

   ```java
   public boolean foo(String[] args) {
       // your code goes here
   }
   ```
3.3 Identifying Errors in Code

1. Where is the error in this code sequence?
   ```java
double[] a = { 3.3, 26.0, 48.4 };
a[4] = 2.5;
```

2. Where is the error in this code sequence?
   ```java
double[] a = { 3.3, 26.0, 48.4 };  
System.out.println(a[-1]);
```

3. Where is the error in this code sequence?
   ```java
double[] a = { 3.3, 26.0, 48.4 };  
System.out.print(a{1});
```

4. Where is the error in this code sequence?
   ```java
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?
   ```java
double a[3] = { 3.3, 26.0, 48.4 };
```

6. Where is the error in this code sequence?
   ```java
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?
   ```java
int[] a = { 3, 26, 48, 5 };  
a.length = 10;
```

8. Where is the logic error in this code sequence?
   ```java
int[] a = { 3, 26, 48, 5 };  
System.out.println("The array elements are " + a);