

JB0240R REVIEW: ARRAYS AND STRINGS*

R.G. (Dick) Baldwin

This work is produced by OpenStax-CNX and licensed under the
Creative Commons Attribution License 3.0[†]

Abstract

This module contains review questions and answers keyed to the module titled Jb0240: Java OOP: Arrays and Strings.

1 Table of Contents

- Preface (p. 1)
- Questions (p. 1)
 - 1 (p. 1) , 2 (p. 1) , 3 (p. 2) , 4 (p. 2) , 5 (p. 2) , 6 (p. 2) , 7 (p. 2) , 8 (p. 2) , 9 (p. 2) , 10 (p. 2) , 11 (p. 3) , 12 (p. 7) , 13 (p. 3) , 14 (p. 3) , 15 (p. 3) , 16 (p. 3) , 17 (p. 3) , 18 (p. 3)
- Listings (p. 4)
- Answers (p. 5)
- Miscellaneous (p. 9)

2 Preface

This module contains review questions and answers keyed to the module titled Jb0240: Java OOP: Arrays and Strings¹.

The questions and the answers are connected by hyperlinks to make it easy for you to navigate from the question to the answer and back again.

3 Questions

3.1 Question 1 .

True or false? Arrays and Strings are true objects. If false, explain why.

Answer 1 (p. 9)

3.2 Question 2

True or false? It is easy to write outside the bounds of a **String** or an array. If false, explain why.

Answer 2 (p. 9)

*Version 1.2: Nov 26, 2012 9:33 am +0000

[†]<http://creativecommons.org/licenses/by/3.0/>

¹<http://cnx.org/content/m45214>

3.3 Question 3

You must declare a variable capable of holding a reference to an array object before you can use it. In declaring the variable, you must provide two important pieces of information. What are they?

Answer 3 (p. 9)

3.4 Question 4

Provide code fragments that illustrate the two different syntaxes that can be used to declare a variable capable of holding a reference to an array object that will store data of type int.

Answer 4 (p. 9)

3.5 Question 5

True or false? When you declare a variable capable of holding a reference to an array object, the memory required to contain the array object is automatically allocated. If false, explain why and show how memory can be allocated.

Answer 5 (p. 8)

3.6 Question 6

True or false? It is required that you simultaneously declare the name of the variable and cause memory to be allocated to contain the array object in a single statement. If false, explain why and show code fragments to illustrate your answer.

Answer 6 (p. 8)

3.7 Question 7

True or false? Array indices always begin with 1. If false, explain why.

Answer 7 (p. 8)

3.8 Question 8

What is the name of the property of arrays that can be accessed to determine the number of elements in the array? Provide a sample code fragment that illustrates the use of this property.

Answer 8 (p. 8)

3.9 Question 9

What types of data can be stored in array objects?

Answer 9 (p. 8)

3.10 Question 10

True or false? Just as in other languages, when you create a multi-dimensional array, the secondary arrays must all be of the same size. If false, explain your answer. Then provide a code fragment that illustrates your answer or refer to a sample program in Jb0240: Java OOP: Arrays and Strings² that illustrates your answer.

Answer 10 (p. 7)

²<http://cnx.org/content/m45214>

3.11 Question 11

True or false? Just as in other languages, when declaring a two-dimensional array, it is necessary to declare the size of the secondary dimension when the array is declared. If false, explain your answer. Then provide a code fragment that illustrates your answer or refer to a sample program in Jb0240: Java OOP: Arrays and Strings ³ that illustrates your answer.

Answer 11 (p. 7)

3.12 Question 12

True or false? Java allows you to assign one array to another. Explain what happens when you do this. Then provide a code fragment that illustrates your answer or refer to a sample program in Jb0240: Java OOP: Arrays and Strings ⁴ that illustrates your answer.

Answer 12 (p. 7)

3.13 Question 13

Give a brief description of the concept of a string and list the names of two classes used to implement strings?

Answer 13 (p. 7)

3.14 Question 14

What is the syntax that is used to create a literal string? Provide a code fragment to illustrate your answer.

Answer 14 (p. 7)

3.15 Question 15

Explain the difference between objects of types **String** and **StringBuffer** .

Answer 15 (p. 7)

3.16 Question 16

Provide a code fragment that illustrates how to concatenate strings.

Answer 16 (p. 6)

3.17 Question 17

Provide a code fragment that declares and instantiates an array object capable of storing references to two **String** objects. Explain what happens when this code fragment is executed. Then show a code fragment that will allocate memory for the actual **String** objects.

Answer 17 (p. 6)

3.18 Question 18

Write a Java application that illustrates the creation and manipulation of a two-dimensional array with the sub arrays being of different lengths. Also cause your application to illustrate that an attempt to access an array element out of bounds results in an exception being thrown. Catch and process the exception. Display a termination message with your name.

Answer 18 (p. 5)

³<http://cnx.org/content/m45214>

⁴<http://cnx.org/content/m45214>

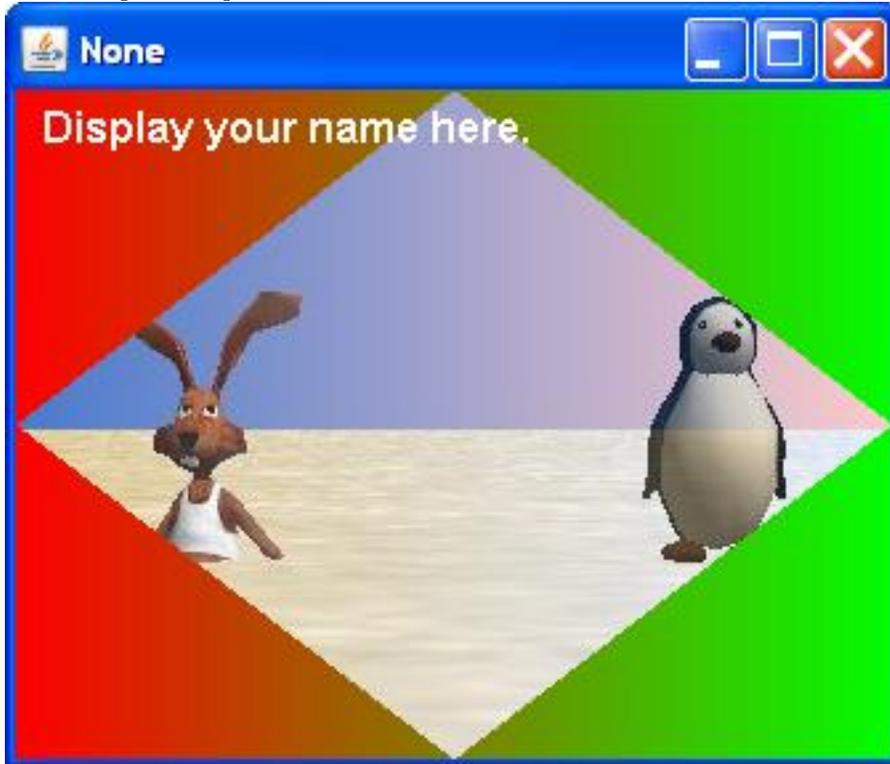
4 Listings

- Listing 1 (p. 5) . Listing for Answer 18.

What is the meaning of the following two images?

This image was inserted here simply to insert some space between the questions and the answers to keep them from being visible on the screen at the same time.

The image is also an example of the kinds of things that we do in my course titled ITSE 2321, Object-Oriented Programming.



This image was also inserted for the purpose of inserting space between the questions and the answers.



5 Answers

5.1 Answer 18

Listing 1: Listing for Answer 18.

```
class SampProg10 { //define the controlling class
public static void main(String[] args){ //define main
//Declare a two-dimensional array with a size of 3 on
// the primary dimension but with different sizes on
// the secondary dimension.
//Secondary size not specified
int[][] myArray = new int[3][];
myArray[0] = new int[2]; //secondary size is 2
myArray[1] = new int[3]; //secondary size is 3
myArray[2] = new int[4]; //secondary size is 4

//Fill the array with data
for(int i = 0; i < 3; i++){
    for(int j = 0; j < myArray[i].length; j++){
        myArray[i][j] = i * j;
    } //end inner loop
} //end outer loop
```

```
//Display data in the array
for(int i = 0; i < 3; i++){
    for(int j = 0; j < myArray[i].length; j++){
        System.out.print(myArray[i][j]);
    }//end inner loop
    System.out.println();
}//end outer loop

//Attempt to access an out-of-bounds array element
try{
    System.out.println(
        "Attempt to access array out of bounds");
    myArray[4][0] = 7;
}catch(ArrayIndexOutOfBoundsException e){
    System.out.println(e);
}//end catch

System.out.println("Terminating, Dick Baldwin");

} //end main
} //End SampProg10 class. Note no semicolon required
```

Back to Question 18 (p. 3)

5.2 Answer 17

The following statement declares and instantiates an array object capable of storing references to two **String** objects.

NOTE:

```
String[] myArrayOfStringReferences = new String[2];
```

Note however, that this array object doesn't contain the actual string data. Rather, it simply sets aside memory for storage of two references to **String** objects. No memory has been set aside to store the characters that make up the individual strings. You must allocate the memory for the actual **String** objects separately using code similar to the following.

NOTE:

```
myArrayOfStringReferences[0] = new String(
    "This is the first string.");
myArrayOfStringReferences[1] = new String(
    "This is the second string.");
```

Back to Question 17 (p. 3)

5.3 Answer 16

Java supports string concatenation using the overloaded + operator as shown in the following code fragment:

NOTE:

```
"My variable has a value of " + myVar +  
" at this point in the program."
```

Back to Question 16 (p. 3)

5.4 Answer 15

String objects cannot be modified once they have been created. **StringBuffer** objects can be modified

Back to Question 15 (p. 3)

5.5 Answer 14

The Java compiler considers a series of characters surrounded by quotation marks to be a literal string, as in the following code fragment:

NOTE:

```
"This is a literal string in Java."
```

Back to Question 14 (p. 3)

5.6 Answer 13

A string is commonly considered to be a sequence of characters stored in memory and accessible as a unit. Java implements strings using the **String** class and the **StringBuffer** class.

Back to Question 13 (p. 3)

5.7 Answer 12

Java allows you to assign one array to another. When you do this, you are simply making another copy of the reference to the same data in memory. Then you have two references to the same data in memory. This is illustrated in the program named **array02.java** in Jb0240: Java OOP: Arrays and Strings ⁵.

Back to Question 12 (p. 3)

5.8 Answer 11

False. When declaring a two-dimensional array, it is not necessary to declare the size of the secondary dimension when the array is declared. Declaration of the size of each sub-array can be deferred until later as illustrated in the program named **array01.java** in Jb0240: Java OOP: Arrays and Strings ⁶.

Back to Question 11 (p. 3)

5.9 Answer 10

False. Java can be used to produce multi-dimensional arrays that can be viewed as an array of arrays. However, the secondary arrays need not all be of the same size. See the program named **array01.java** in Jb0240: Java OOP: Arrays and Strings ⁷.

Back to Question 10 (p. 2)

⁵<http://cnx.org/content/m45214>

⁶<http://cnx.org/content/m45214>

⁷<http://cnx.org/content/m45214>

5.10 Answer 9

Array objects can contain any Java data type including primitive values, references to ordinary objects, and references to other array objects.

Back to Question 9 (p. 2)

5.11 Answer 8

All array objects have a **length** property that can be accessed to determine the number of elements in the array as shown below.

NOTE:

```
for(int cnt = 0; cnt < myArray.length; cnt++)
    myArray[cnt] = cnt;
```

Back to Question 8 (p. 2)

5.12 Answer 7

False. Array indices always begin with 0.

Back to Question 7 (p. 2)

5.13 Answer 6

False. While it is possible to simultaneously declare the name of the variable and cause memory to be allocated to contain the array object, it is not necessary to combine these two processes. You can execute one statement to declare the variable and another statement to cause the memory for the array object to be allocated as shown below.

NOTE:

```
int[] myArray;
.
.
.
myArray = new int[25];
```

Back to Question 6 (p. 2)

5.14 Answer 5

False. As with other objects, the declaration of the variable does not allocate memory to contain the array object. Rather it simply allocates memory to contain a reference to the array object. Memory to contain the array object must be allocated from dynamic memory using statements such as the following.

NOTE:

```
int[] myArray = new int[15];
int myArray[] = new int[25];
int[] myArray = {1,2,3,4,5}
```

Back to Question 5 (p. 2)

5.15 Answer 4

NOTE:

```
int[] myArray;  
int myArray[];
```

Back to Question 4 (p. 2)

5.16 Answer 3

In declaring the variable, you must provide two important pieces of information:

- the name of the variable
- the type of the variable, which indicates the type of data to be stored in the array

Back to Question 3 (p. 2)

5.17 Answer 2

False. Java has a true array type and a true **String** type with protective features to prevent your program from writing outside the memory bounds of the array or the **String**.

Back to Question 2 (p. 1)

5.18 Answer 1

True.

Back to Question 1 (p. 1)

6 Miscellaneous

This section contains a variety of miscellaneous information.

NOTE: **Housekeeping material**

- Module name: Jb0240r Review: Arrays and Strings
- File: Jb0240r.htm
- Originally published: 1997
- Published at cnx.org: November 26, 2012

NOTE: **Disclaimers: Financial** : Although the Connexions site makes it possible for you to download a PDF file for this module at no charge, and also makes it possible for you to purchase a pre-printed version of the PDF file, you should be aware that some of the HTML elements in this module may not translate well into PDF.

I also want you to know that, I receive no financial compensation from the Connexions website even if you purchase the PDF version of the module.

In the past, unknown individuals have copied my modules from cnx.org, converted them to Kindle books, and placed them for sale on Amazon.com showing me as the author. I neither receive compensation for those sales nor do I know who does receive compensation. If you purchase such a book, please be aware that it is a copy of a module that is freely available on cnx.org and that it was made and published without my prior knowledge.

Affiliation : I am a professor of Computer Information Technology at Austin Community College in Austin, TX.

-end-