**Chapter 7: Arrays**

**Multiple Choice Questions**:

1) In Java, array indexes always begin at \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ .

a) -1

b) 0

c) 1

d) 2

e) you can declare an array to have any indexes you choose

Answer: b

Explanation: In Java, the array indexes are from 0 to one less than the length of the array.

2) Which of the following statements best describes this line of code?

numbers[5] = 12;

a) The value 12 is put into the numbers array at the location with index 5.

b) The value 5 is put into the numbers array at the location with index 12.

c) numbers is declared to be an array of 5 elements, each of which contains the number 12.

d) numbers is declared to be an array of 12 elements, each of which contains the number 5.

e) none of the above is correct

Answer: a

Explanation: The code represents the assignment of the value 12 to the location in the numbers array with index 5.

3) Which of the statements is true about the following code snippet?

int[] array = new int[25];

array[25] = 2;

a) The integer value 2 will be assigned to the last index in the array.

b) The integer value 25 will be assigned to the second index in the array.

c) The integer value 25 will be assigned to the third value in the array.

d) This code will result in a compile-time error.

e) This code will result in a run-time error.

Answer: e

Explanation: This code will throw an ArrayIndexOutOfBoundsException, since the last index in this array will be 24. This causes a run-time error.

4) What does this array contain?

String [] studentNames = new String[25];

a) names

b) students

c) studentNames

d) Strings

e) references to Strings

Answer: e

Explanation: The elements of the array are Strings, which are objects. As such, the array contains references to objects, not the objects themselves. The Strings may be the names of students, but the array elements are not names, students, or studentNames.

5) Which of the following array declarations are invalid?

a) int[] grades = new int[5];

b) int grades[] = new int[5];

c) int[] grades = { 91, 83, 42, 100, 77 };

d) all of the above are valid

e) none of the above are valid

Answer: d

Explanation: All three of these are valid array declarations. Choice b uses an alternate syntax. Choice c uses an initializer list to initialize the array.

6) Which of the following is a true statement?

a) Arrays are passed as parameters to methods like primitive types.

b) Arrays are passed as parameters to methods like object types.

c) Arrays cannot be passed as parameters to methods.

d) All of the above are true.

e) None of the above are true.

Answer: b

Explanation: Arrays are passed to methods by reference. This means that if the content of the array is changed in a method, the change will be reflected in the calling method.

7) Suppose we have an array of String objects identified by the variable names. Which of the following for loops will *not* correctly process each element in the array.

a) for(int i = 0; i < names.length; i++)

b) for(String name : names)

c) for(int i = 0; i < names.length(); i++)

d) none of these will correctly process each element

e) all of these will correctly process each element

Answer: c

Explanation: Choice c will not process each element correctly due to a syntax error. The length constant is not a method and, therefore, does not have parentheses after it. Choice b is an example of using a *foreach* loop to process an array, and choice a is a correct for loop.

8) Which of the following statements will assign the first command-line argument sent into a Java program to a variable called argument?

a) argument = System.getFirstArgument();

b) argument = System.getArgument[1];

c) argument = System.getArgument[0];

d) argument = args[0];

e) argument = args[1];

Answer: d

Explanation: Choice d is the correct answer. The System object does not have any methods called getFirstArgument or getArgument, and the args array's index starts at 0. Therefore the other choices are incorrect.

9) Which of the following method declarations correctly defines a method with a variable length parameter list?

a) public int average(int[] list)

b) public int average(int ... list)

c) public int average(...)

d) public int average(int a, int b, int c, ...)

e) public int average(integers)

Answer: b

Explanation: The only choices with valid syntax are choice a and choice b. Choice a represents a method declaration with a single parameter, which is a reference to an array. Choice b correctly represents a valid declaration for a method with a variable length parameter list.

10) Which of the following is a valid declaration for a two-dimensional array?

a) int[][] matrix;

b) int[2] matrix;

c) int[]\*\* matrix;

d) int[] matrix;

e) none of these are correct

Answer: a

Explanation: Choice a is the only valid declaration for a two-dimensional array. Choices b and c contain invalid Java syntax, and choice d is a valid declaration for a single dimensional array.

11) Which of the following lines of code accesses the second element of the first array in a two-dimensional array of integers, numbers, and stores the result in a variable called num?

a) num = numbers[1][2];

b) num = numbers[0][1];

c) num = numbers.getElement(1, 2);

d) num = numbers.getElement(0, 1);

e) none of the above are correct

Answer: b

Explanation: Choice b accesses the second element of the first array. Choice a accesses the third element of the second array. Choices c and d do not represent valid Java syntax.

12) Which of the following are true about two-dimensional arrays?

a) Two-dimensional integer arrays cannot be initialized via an initializer list.

b) Two-dimensional arrays can only store up to 10 elements per array (or per row).

c) Two-dimensional arrays are not accessible using for loops.

d) Two-dimensional arrays cannot hold objects.

e) None of the above is true.

Answer: e

Explanation: None of the statements about two-dimensional arrays are true.

13) How do you determine the number of command-line arguments a program has?

a) The numArgs constant contains the number of command-line arguments.

b) The first member of the args array is the count of the number of command-line arguments.

c) The number of command-line arguments is the length of the args array.

d) The number of command-line arguments is one less than the length of the args array.

e) The number of command-line arguments is fixed and is 5.

Answer: c

Explanation: args is the array that contains the command-line arguments as Strings. args.length is the number of command-line arguments.

14) What is the limit of the number of variable parameters that can be passed to a method that has a variable-length parameter list?

a) 5

b) 10

c) 16

d) less than 100

e) none of the above

Answer: e

Explanation: A variable-length parameter list is converted to an array when the method is called. Since the length of an array is not limited to the numbers 5, 10, 16 or some number less than 100, none of the choices a) through d) can be correct.

15) Multi-dimensional arrays that contain arrays of different lengths in any one dimension are called \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

a) ragged arrays

b) static arrays

c) two-dimensional arrays

d) constant arrays

e) overloaded arrays

Answer: a

Explanation: Ragged arrays are multi-dimensional arrays that contain arrays at the same dimension with differing lengths.**True/False Questions**:

1) In Java, array indexes begin at 0 and end at one less than the length of the array.

Answer: True

Explanation: For an array of length n, the indexes begin at 0 and end at n – 1 in Java.

2) If an array is declared to hold objects, none of the objects can have instance variables that are arrays.

Answer: False

Explanation: If an array holds objects, each element of the array is a reference to an object. There are no restrictions on the instance variables or the methods of the objects that are the referenced by the elements of the array.

3) An array declared as an int[] can contain elements of different primitive types.

Answer: False

Explanation: An array that has been declared with a specific type may only contain elements of that type. In this case the array can only contain integers.

4) The elements of a two-dimensional array are called rows and columns.

Answer: False

Explanation: We consider the elements of a two-dimensional array as being arranged in a table that has rows and columns. The rows are represented by the first subscript of the array, while the column within a row is indicated by the second subscript. The element in a particular row and column of a two-dimensional array is still called an element.

5) It is possible to store 11 elements in an array that is declared in the following way.

int[] array = new int[10];

Answer: False

Explanation: An array declared as above can only store 10 elements.

6) If a program attempts to access an element outside of the range of the array indexes, a run-time error will occur.

Answer: True

Explanation: If a program attempts to access an element outside of the range of the array indexes, an ArrayOutOfBoundsException will be thrown at run-time.

7) An array cannot hold object types.

Answer: False

Explanation: An array can be declared to hold references to objects.

8) It is possible to send in data to a Java program via the command-line.

Answer: True

Explanation: Command-line arguments can be sent in to a Java program. They are sent into the program via the args[] array.

9) It is possible for a method to have a variable length parameter list, meaning that the method can take in any number of parameters of a specified data type.

Answer: True

Explanation: Java supports variable length parameter lists for methods.

10) In Java it is not possible to have arrays of more than two dimensions.

Answer: False

Explanation: It is possible to have arrays of any dimension in Java. **Short Answer Questions**:

1) Explain how arrays are passed to methods as parameters.

Answer: Arrays are passed to methods by reference. This means that a reference to the original array is sent into the method. Therefore any changes that are made to the array in the method will be reflected in the original array in the calling method.

2) Write the declaration for an array of doubles called averages that is initialized with an initializer list.

Answer:

double[] averages = { 25.2, 36.18, 42.1, 30.5 };

3) Write the declaration for a two-dimensional array of integers that can be thought of as a table with three rows and three columns. Assign the value 3 to the cell that is in the second row and the third column.

Answer:

int[][] table = new int[3][3];

table[1][2] = 3;

4) Write a loop that cycles through an array of String objects called names and prints them out, one per line.

Answer: using a *for* loop:

for(int i = 0; i < names.length; i++)

System.out.println(names[i]);

Answer:: using a *foreach* loop:

for(String n : names)

System.out.println(n);

5) How do you determine how the elements of an array can be used in a program?

Answer:

Refer to the array declaration to determine how the elements can be used. For example, if the array is declared with the type int [], then each element of the array is an individual int, and can be used in any way that is appropriate for an int. If the array is declared with a type that refers to a class, then the array elements are all references to objects of that class and can be used as such.

6) Student is a class that defines data fields and methods for an individual student. Write the declaration of an array named roster that can be used to reference 24 Student objects.

Answer:

Student [] roster = new Student[24];

7) Write a method called doubleSize that accepts an integer array as a parameter and returns a reference to a new integer array that is twice as long and contains all of the elements of the first array in the same positions.

Answer:

public int[] doubleSize(int[] originalArray) {

int[] newArray = int[originalArray.length\*2];

for(int i = 0; i < originalArray.length; i++)

newArray[i] = originalArray[i];

return newArray;

}

8) Circle is a class that has data and methods related to circles. How many Circle objects are created by the following declaration?

Circle [] shapes = new Circle[12];

Answer: No Circle objects are created by the declaration. The array declaration creates references to 12 Circles, but the Circle objects must be separately instantiated and assigned to the array members.

9) What is the purpose of command-line arguments to a Java program? How can they be used?

Answer: Command-line arguments are passed to the main method when execution commences. They can be used to provide input to a program in a non-interactive way. They are passed as an array of Strings. Individual arguments can be accessed using array subscript notation.

10) Write a method that takes in an arbitrary number of String objects, and then prints out all of them that have over 10 characters.

Answer:

public void printLongStrings(String ... words) {

for(String w : words)

if(w.length() > 10)

System.out.println(w);

}

11) Write a method that takes in at least one integer and returns the largest of all integer parameters sent in.

Answer:

public void largest(int first, int ... numbers) {

int currentLargest = first;

for(int num : numbers)

if(num > currentLargest)

currentLargest = num;

return currentLargest;

}

12) Write a method that accepts an array of integers as a parameter and returns a reference to an array that contains the even numbers in the array original array. The returned array should have a size equal to the number of even numbers in the original array.

Answer:

public int[] getEvenArray(int[] numbers) {

int size = 0;

for(int i = 0; i < numbers.length; i++)

if(numbers[i]%2 == 0)

size++;

int[] evenArray = new int[size];

int evenArrayIndex = 0;

for(int i = 0; i < numbers.length; i++) {

if(numbers[i]%2 == 0) {

evenArray[evenArrayIndex] = numbers[i];

evenArrayIndex++;

}//end if

}//end for

}

13) Write a short program that accepts an arbitrary number of command line arguments, and prints out those containing the character 'z'.

Answer:

public class PrintZArgs {

public static void main(String[] args) {

for(String s : args) {

boolean printed = false;

for(int i = 0; (i < s.length) && (!printed); i++)

if(s.charAt(i) == 'z') {

printed = true;

System.out.println(s);

} // end if

// end inner for loop

}//end foreach

}//end main

}//end class

14) Write a line of code that initializes a two-dimensional array of integers using an initializer list.

Answer:

int[][] numbers = { {2,3},{4,5} };

15) The following declaration allocates space for how many double variables?

double [][][] temperatures = new double [10][20[]30];

Answer: This is the declaration of a 3-dimensional array. It can be viewed as a 10-element array (the first subscript) where each array element is a 2-dimensional array. The 2-dimensional arrays have 20 rows and 30 columns, so they have 20 x 30 = 600 elements. There are 10 2-dimensional arrays, so the total number of doubles allocated by this declaration is 10 x 600 = 6000 doubles.