CSE110: Principles of Programming
Lecture 8: Arrays and ArrayLists

Name:
ID:
Program:

Section 1:

Q1: Which of the following statements about arrays are *true*?

A. An array is a group of variables containing values that all have the same type.

B. Elements are located by index or subscript.

C. The length of an array c is determined by the expression c.length();.

D. The zeroth element of array c is specified by c[0].

a. A, C, D.

b. A, B, D.

c. C, D.

d. A, B, C, D.

Q2: Consider the array:

s[0] = 7
s[1] = 0
s[2] = -12
s[3] = 9
s[4] = 10
s[5] = 3
s[6] = 6

The value of s[s[6] - s[5]] is:

a. 0.

b. 3.

c. 9.

d. 0.

Section 2:

Q1: A programmer must do the following before using an array:

a. declare then reference the array.

b. create then declare the array.

c. create then reference the array.

d. declare then create the array.

Q2: Consider the code segment below. Which of the following statements is *false*?

```
int[] g;
g = new int[ 23 ];
```

a. The first statement declares an array reference.
b. The second statement creates the array.
c. g is a reference to an array of integers.
d. The value of g[3] is -1.

Section 3:

Q1: Which of the following statements about creating arrays and initializing their elements is *false*?
a. The new keyword should be used to create an array.
b. When an array is created, the number of elements must be placed in square brackets following the type of element being stored.
c. The elements of an array of integers have a value of null before they are initialized.
d. A for loop is commonly used to set the values of the elements of an array.

Q2: What do the following statements do?

```
double[] array;
array = new double[ 14 ];
```

a. Create a double array containing 13 elements.
b. Create a double array containing 14 elements.
c. Create a double array containing 15 elements.
d. Declare but do not create a double array.

Q3: Which of the following initializer lists would correctly set the elements of array n?
a. int[] n = { 1, 2, 3, 4, 5 };.
b. array n[int] = { 1, 2, 3, 4, 5 };.
c. int n[5] = { 1; 2; 3; 4; 5 };.
d. int n = new int(1, 2, 3, 4, 5);.

7.4 Q4: Constant variables also are called ____.
a. write-only variables.

b. finals.
c. named constants.
d. All of the above.

Q5: Which of the following will *not* produce a compiler error?
a. Changing the value of a constant after it is declared.
b. Changing the value at a given index of an array after it is created.
c. Using a final variable before it is initialized.
d. All of the above will produce compiler errors.

Q6: Consider the program below:

```
public class Test
{
  public static void main( String[] args )
  {
    int[] a;
    a = new int[ 10 ];

    for ( int i = 0; i < a.length; i++ )
      a[ i ] = i + 2;
    int result = 0;
    for ( int i = 0; i < a.length; i++ )
      result += a[ i ];
    System.out.printf( "Result is: %d\n", result );
  } // end main
} // end class Test
```

The output of this program will be:
a. Result is: 62.
b. Result is: 64.
c. Result is: 65.
d. Result is: 67.

Q7: Consider the class below:

```
public class Test
{
  public static void main( String[] args )
  {
    int[] a = { 99, 22, 11, 3, 11, 55, 44, 88, 2, -3 };

    int result = 0;
```

```
      for ( int i = 0; i < a.length; i++ )
      {
         if ( a[ i ] > 30 )
            result += a[ i ];
      } // end for

      System.out.printf( "Result is: %d\n", result );
   } // end main
} // end class Test
```

The output of this Java program will be:
a. Result is: 280.
b. Result is: 154.
c. Result is: 286.
d. Result is: 332.

Q8: Which flag in a format specifier indicates that values with fewer digits than the field width should begin with a leading 0?
a. p.
b. l.
c. w.
d. 0.

Q9: Invalid possibilities for array indices include _____.
a. Positive integers.
b. Negative integers.
c. Zero.
d. None of the above.

Q10: Which expression adds 1 to the element of array arrayName at index i?
a. ++arrayName[i].
b. arrayName++[i].
c. arrayName[i++].
d. None of the above.

Q11: Attempting to access an array element out of the bounds of an array, causes a(n) _____.
a. ArrayOutOfBoundsException.

b. ArrayElementOutOfBoundsException.
c. ArrayIndexOutOfBoundsException.
d. ArrayException.

Q12: Which of the following statements is *false*?
a. An exception indicates a problem that occurs while a program executes.
b. Exception handling enables you to create fault-tolerant programs that can resolve (or handle) exceptions—in many cases, this allows a program to continue executing as if no problems were encountered.
c. The catch block contains the code that might throw an exception, and the try block contains the code that handles the exception if one occurs.
d. Inside the catch block, you can use the parameter’s identifier to interact with a caught exception object.

Section 4

Q1: Consider integer array values, which contains 5 elements. Which statements successfully swap the contents of the array at index 3 and index 4?

a.
```
values[ 3 ] = values[ 4 ];
values[ 4 ] = values[ 3 ];
```
b.
```
values[ 4 ] = values[ 3 ];
values[ 3 ] = values[ 4 ];
```
c.
```
int temp = values[ 3 ];
values[ 3 ] = values[ 4 ];
values[ 4 ] = temp;
```
d.
```
int temp = values[ 3 ];
values[ 3 ] = values[ 4 ];
values[ 4 ] = values[ 3 ];
```
Q2: In this question, assume a class, Book, has been defined. Which set of statements creates an array of Book objects?

a.
```
Book[] books;
books = new Book[ numberElements ];
```
b.
```
Book[] books];
books = new Book()[ numberElements ];
```

c.

```
new Book() books[];
books = new Book[ numberElements ];
```

d. All of the above.

Section 5

Q1: Assume array items contains the integer values 0, 2, 4, 6 and 8. Which of the following set of statements uses the enhanced for loop to display each value in array items?

a.

```
for ( int i = 0; i < items.length; i++ )
  System.out.prinf( "%d\n", items[ i ] );
```

b.

```
for ( int i : items )
  System.out.prinf( "%d\n", items[ i ] );
```

c.

```
for ( int i : items )
  System.out.prinf( "%d\n", i );
```

d.

```
for ( int i = 0 : items.length )
  System.out.prinf( "%d\n", items[ i ] );
```

Q2: Which of the following tasks *cannot* be performed using an enhanced for loop?

a. Calculating the product of all the values in an array.
b. Displaying all even element values in an array.
c. Comparing the elements in an array to a specific value.
d. Incrementing the value stored in each element of the array.

Section 6

Q1: Which statement correctly passes the array items to method takeArray? Array items contains 10 elements.

a. takeArray(items[]).
b. takeArray(items).
c. takeArray(items[9]).
d. Arrays cannot be passed to methods—each item must be sent to the method separately.

Q2: Consider array items, which contains the values 0, 2, 4, 6 and 8. If method changeArray is called with the method call changeArray(items,

items[2]), what values are stored in items after the method has finished executing?

```
public static void changeArray( int[] passedArray, int value )
{
  passedArray[ value ] = 12;
  value = 5;
} // end method changeArray
```

a. 0, 2, 5, 6, 12.
b. 0, 2, 12, 6, 8.
c. 0, 2, 4, 6, 5.
d. 0, 2, 4, 6, 12.

Q3: When an argument is passed by reference:
a. a copy of the argument's value is passed to the called method.
b. changes to the argument do not affect the original variable's value in the caller.
c. the called method can access the argument's value in the caller directly and modify that data.
d. the original value is removed from memory.
Ans: c. the called method can access the argument's value in the caller directly and modify that data.

Section 7

Q1: What kind of application tests a class by creating an object of that class and calling the class's methods?
a. Pseudo application.
b. Debugger.
c. Tester.
d. Test harness.

Section 8

Q1: In Java, multidimensional arrays:
a. are not directly supported.
b. are implemented as arrays of arrays.
c. are often used to represent tables of values.
d. All of the above.

Q2: In array items, which expression below accesses the value at row 3 and column 4?
a. items[3].[4].

b. items[3[4]].
c. items[3][4].
d. items[3, 4].

Q3: An array with *m* rows and *n* columns is *not*:
A. An *m*-by-*n* array.
B. An *n*-by-*m* array.
C. A two-dimensional array.
D. A dual-transcripted array.

a. A and C.
b. A and D.
c. B and D.
d. B and C.

Q4: Which statement below initializes array items to contain 3 rows and 2 columns?
a. int[][] items = { { 2, 4 }, { 6, 8 }, { 10, 12 } };.
b. int[][] items = { { 2, 6, 10 }, { 4, 8, 12 } };.
c. int[][] items = { 2, 4 }, { 6, 8 }, { 10, 12 };.
d. int[][] items = { 2, 6, 10 }, { 4, 8, 12 };.

Q5: For the array in the previous question, what is the value returned by items[1][0]?
a. 4.
b. 8.
c. 12.
d. 6.

Q6: Which of the following statements creates a multidimensional array with 3 rows, where the first row contains 1 element, the second row contains 4 elements and the final row contains 2 elements?
a. int[][] items = { { 1, null, null, null }, { 2, 3, 4, 5 }, { 6, 7, null, null } };.
b. int[][] items = { { 1 }, { 2, 3, 4, 5 }, { 6, 7 } };.
c. int[][] items = { { 1 }, { 2, 3, 4, 5 }, { 6, 7 }, {});.
d. int[][] items = { { 1 }, { 4 }, { 2 } };.

Q7: Which of the following sets of statements creates a multidimensional array with 3 rows, where the first row contains 1 value, the second row contains 4 items and the final row contains 2 items?

a.

```
int[][] items;
items = new int[ 3 ][ ? ];
items[ 0 ] = new int[ 1 ];
items[ 1 ] = new int[ 4 ];
items[ 2 ] = new int[ 2 ];
```

b.

```
int[][] items;
items = new int[ 3 ][ ];
items[ 0 ] = new int[ 1 ];
items[ 1 ] = new int[ 4 ];
items[ 2 ] = new int[ 2 ];
```

c.

```
int[][] items;
items = new int[ ? ][ ? ];
items[ 0 ] = new int[ 1 ];
items[ 1 ] = new int[ 4 ];
items[ 2 ] = new int[ 2 ];
```

d.

```
int[][] items;
items[ 0 ] = new int[ 1 ];
items[ 1 ] = new int[ 4 ];
items[ 2 ] = new int[ 2 ];
```

Q8: The preferred way to traverse a two-dimensional array is to use _____.

a. a do while statement.

b. a for statement.

c. two nested for statements.

d. three nested for statements.

Q9: Which set of statements totals the items in each row of two-dimensional array items, and displays each total?

a.

```
int total = 0;

for ( int row = 0; row < items.length; row++ )
{
  total = 0;
  for ( int column = 0; column < a[ row ].length; column++ )
    total += a[ row ][ column ];

  System.out.printf( "%d\n", total );
}
```

b.

```
int total = 0;

for ( int row = 0; row < items.length; row++ )
{
  for ( int column = 0; column < a[ row ].length; column++ )
    total += a[ row ][ column ];

  System.out.printf( "%d\n", total );
}
```

c.

```
int total = 0;

for ( int row = 0; row < items.length; row++ )
{
  for ( int column = 0; column < a[ column ].length; column++ )
    total += a[ row ][ column ];

  System.out.printf( "%d\n", total );
}
```

d.

```
int total = 0;

for ( int row = 0; row < items.length; row++ )
{
  total = 0;

  for ( int column = 0; column < a[ column ].length; column++ )
    total += a[ row ][ column ];

  System.out.printf( "%d\n", total );
}
```

Section 8

Q1: Which set of statements totals the values in two-dimensional int array items?

a.

```
int total = 0;

for ( int subItems : items )
   for ( int item : subItems )
      total += item;
```

b.

```
int total = 0;

for ( int item: int[] subItems : items )
   total += item;
```

c.

```
int total = 0;

for ( int[] subItems : items )
   for ( int item : items )
      total += item;
```

d.

```
int total = 0;

for ( int[] subItems : items )
   for ( int item : subItems )
      total += item;
```

Section 9

Q1: An argument type followed by a(n) ____ in a method’s parameter list indicates that the method receives a variable number of arguments of that particular type.

a. square brackets ([]).

b. ellipsis (…).

c. varargs keyword.

d. All of the above are acceptable to indicate a variable number of arguments.

Section 10

Q1: Which command below runs TestProgram, and passes in the values files.txt and 3?

a. java TestProgram files.txt 3.

b. java TestProgram files.txt, 3.

c. java TestProgram "files.txt", "3".
d. java TestProgram (the arguments files.txt and 3 were passed in when the application was compiled).

Q2: Which method call converts the value in variable stringVariable to an integer?
a. Convert.toInt(stringVariable).
b. Convert.parseInt(stringVariable).
c. Integer.parseInt(stringVariable).
d. Integer.toInt(stringVariable).

Section 11
Q1: Class Arrays methods sort, binarySearch, equals and fill are overloaded for primitive-type arrays and Object arrays. In addition, methods __________ and __________ are overloaded with generic versions.
a. sort, binarySearch.
b. sort, fill.
c. binarySearch, equals.
d. binarySearch, fill.

7.13 Q2: Class Arrays provides method __________ for comparing arrays.
a. compare.
b. compares.
c. equal.
d. equals.