

AP Computer Science Final 2011

Part I: Vocabulary (10%)

A/An _____ can hold multiple values of the same data type _____.
The number inside the brackets is the array's _____. It indicates the number of elements, or values, the array can hold. An array's _____ must be a _____ integer expression. It can be a literal value or a variable.
Although an array has only one name, the elements in the array may be accessed and used as individual variables. This is possible because each element is assigned a number known as a _____. A _____ is used as a/an _____ to pinpoint a specific element within an array. The first _____ is assigned the _____ 0, the second element is assigned 1, and so forth.

Word Bank: array, size declarator, subscript, index, simultaneously, non-negative, element

Part II: Multiple Choice (30%)

Question 1

In an array declaration, this indicates the number of elements that the array will have.

Choose one answer.

- a. subscript
- b. size declarator
- c. element sum
- d. reference variable

Question 2

Each element of an array is accessed by a number known as an _____.

Choose one answer.

- a. subscript
- b. size declarator
- c. address
- d. specifier

Question 3

The first subscript in an array is always _____.

Choose one answer.

- a. 1
- b. 0
- c. -1
- d. 1 less than the number of elements

Question 4

The last subscript in an array is always _____.

Choose one answer.

- a. 100
- b. 0
- c. -1
- d. 1 less than the number of elements

Question 5

Array bounds checking happens _____.

Choose one answer.

- a. when the program is compiled
- b. when the program is saved
- c. when the program runs
- d. when the program is loaded into memory

Question 6

This array field holds the number of elements that the array has.

Choose one answer.

- a. size
- b. elements
- c. length
- d. width

Question 7

This search algorithm steps through an array, comparing each item with the search value.

Choose one answer.

- a. binary search
- b. sequential search
- c. selection search
- d. iterative search

Question 8

When initializing a two-dimensional array, you enclose each row's initialization list in _____.

Choose one answer.

- a. braces
- b. parentheses
- c. brackets
- d. quotation marks

Question 9

The statement: `double[] array1 = new double[10]`

Choose one answer.

- a. Declares array1 to be a reference to an array of double values
- b. Creates an instance of an array of 10 double values
- c. Will allow valid subscripts in the range of 0–9
- d. All of the above

Question 10

What will be the value of x[8] after the following code has been executed?

```
final int SUB = 12;
int[] x = new int[SUB];
int y = 100;
for(int i = 0; i < SUB; i++)
{
    x[i] = y;
    y += 10;
}
```

Choose one answer.

- a. 170
- b. 180
- c. 190
- d. 200

Question 11

What will be the results of the following code?

```
final int ARRAY_SIZE = 5;
float[] x = new float[ARRAY_SIZE];
for(int i = 1; i <= ARRAY_SIZE; i++)
{
    x[i] = 10.0;
}
```

Choose one answer.

- a. All the values in the array are initialized to 10.0
- b. All the values, except the first, are set to 10.0
- c. The program will crash when it is executed
- d. There will be a compilation error

Question 12

What would be the results of the following code?

```
int[] x = {55, 33, 88, 22, 99, 11, 44, 66, 77};
int a = 10;
if(x[2] > x[5])
    a = 5;
else
    a = 8;
```

Choose one answer.

- a. a = 5
- b. a = 8
- c. a = 10

d. This is a compilation error, you cannot compare array elements

Question 13

What would be the results after the following code was executed?

```
int[] x = {23, 55, 83, 19};
int[] y = {36, 78, 12, 24};
for(int a = 0; a < x.length; a++)
{
x[a] = y[a];
y[a] = x[a];
}
```

Choose one answer.

- a. y[] = {23, 55, 83, 19} and x[] = {36, 78, 12, 24}
- b. x[] = {36, 78, 12, 24} and y[] = {36, 78, 12, 24}
- c. x[] = {23, 55, 83, 19} and y[] = {23, 55, 83, 19}
- d. This is a compilation error

Question 14

What will be the value for x[1] after the following code is executed?

```
int[] x = {22, 33, 44};
arrayProcess(x);
...
public static void arrayProcess(int[] a)
{
for(int k = 0; k < 3; k++)
{
a[k] = a[k] + 5;
}
}
```

Choose one answer.

- a. 27
- b. 33
- c. 28
- d. 39

Question 15

What would be the results of the following code?

```
int[] array1 = new int[25];
... // Code that will put values in array1
int value = array1[0];
for (int a = 1; a < array1.length; a++)
{
if (array1[a] < value)
value = array1[a];
}
```

}

Choose one answer.

- a. value contains the highest value in array1
- b. value contains the lowest value in array1
- c. value contains the sum of all the values in array1
- d. value contains the average of the values in array1

Question 16

A partially-filled array is normally used

Choose one answer.

- a. When you know how many elements will be in the array
- b. With an accompanying parallel array
- c. With an accompanying integer value that holds the number of items stored in the array
- d. To display array elements

Question 17

In memory, an array of String objects

Choose one answer.

- a. Consists of an array of references to String objects
- b. Is arranged the same as an array of primitive objects
- c. Consists of elements, each of which is a String
- d. Must be initialized when the array is declared

Question 18

Given `String[] str` has been initialized, to convert all the characters in the `String str[0]` to upper case, use the following statement:

Choose one answer.

- a. `str.uppercase();`
- b. `str[0].upperCase();`
- c. `str.toUpperCase();`
- d. `str[0].toUpperCase();`

Question 19

When an array of objects is declared, but not initialized, the array values are set to null.

Answer:

True False

Question 20

The sequential search algorithm

Choose one answer.

- a. Requires the array to be ordered
- b. Returns the value it was searching for

- c. Uses a loop to sequentially step through an array, starting with the first element
- d. Will not execute, if the element is not in the array

Part III: Complete the following Programa (20 %)
Program 1

```
/**
 * Rainfall Class
 * Chapter 8, Programming Challenge 1
 */

public class Rainfall
{
    private double[] rain; // Array to hold rainfall data

    /**
     * Constructor
     * @param r An array of rainfall figures.
     */

    public Rainfall(double r[])
    {
        // Create a new array.
        rain = new double[r.length];

        // Copy the argument's elements to the
        // new array.
        for (int i = 0; i < r.length; i++)
            rain[i] = r[i];
    }

    /**
     * The getTotalRainFall method calculates the total
     * rainfall.
     * @return The total amount of rainfall.
     */

    public double getTotalRainFall()
    {
        double total = 0.0; // Accumulator

        // Accumulate the sum of the rain array elements.
        for (int i = 0; i < rain.length; i++)
            total += rain[i];
    }
}
```

```

    // Return the sum.
    return total;
}

/**
 * The getAverageRainfall method calculates the
 * average amount of rainfall.
 * @return The average amount of rainfall.
 */

public double getAverageRainFall()
{
    return getTotalRainFall() / rain.length;
}

/**
 * The getHighestMonth method determines the month
 * with the highest amount of rainfall.
 * @return The number of the month with the highest
 * amount of rainfall.
 */

public int getHighestMonth()
{
    int highest = 0;

    // Find the element with the highest value.
    for (int i = 1; i < rain.length; i++)
    {
        if (rain[i] > rain[highest])
            highest = i;
    }
    // Return the element number.
    return highest;
}

/**
 * The getLowstMonth method determines the month
 * with the lowest amount of rainfall.
 * @return The number of the month with the lowest
 * amount of rainfall.
 */

public int getLowestMonth()
{
    Complete it...
}

```

```

    }

    /**
     The getRainAt method returns a specified value
     in the array.
     @param e The element number to return.
     @return The value stored in the specified element.
     */

    public double getRainAt(int e)
    {
        return rain[e];
    }
}

```

Program 2

```

import java.util.Scanner;
import java.text.DecimalFormat;

/**
 This program demonstrates a two-dimensional array.
 */

public class CorpSales
{
    public static void main(String[] args)
    {
        final int DIVS = 3; // Three divisions in the company
        final int QTRS = 4; // Four quarters
        double totalSales = 0.0; // Accumulator

        // Create an array to hold the sales for each
        // division, for each quarter.
        double[][] sales = new double[DIVS][QTRS];

        // Create a Scanner object for keyboard input.
        Scanner keyboard = new Scanner(System.in);

        // Display an introduction.
        System.out.println("This program will calculate the " +
            "total sales of");
    }
}

```



```
System.out.println("all the company's divisions. " +  
    "Enter the following sales data:");
```

```
// Nested loops to fill the array with quarterly  
// sales figures for each division.
```

```
for (int div = 0; div < DIVS; div++)  
{  
    for (int qtr = 0; qtr < QTRS; qtr++)  
    {  
        System.out.print("Division " + (div + 1) +  
            ", Quarter " + (qtr + 1) +  
            ": $");  
        sales[div][qtr] = keyboard.nextDouble();  
    }  
    System.out.println(); // Print blank line.  
}
```

```
// Nested loops to add all the elements of the array.
```

Complete it...

```
// Create a DecimalFormat object to format output.
```

```
DecimalFormat dollar = new DecimalFormat("#,##0.00");
```

```
// Display the total sales.
```

```
System.out.println("The total sales for the company " +  
    "are $" + dollar.format(totalSales));
```

```
}  
}
```

Part IV: Program Challenge (40%)

Program Rubric

<http://mail.sad60.k12.me.us/~ysun/Programming%20Rubric>

(C Level & B Level)

Use a 2-D array to write a program that has 3 students and 4 test scores. Ask the user to enter each student's test scores. Display each student's average score like this:

```
This program will calculate the the average of
three students' four tests' scores. Enter the following test scores:
Student 1, Test 1: 90
Student 1, Test 2: 89
Student 1, Test 3: 78
Student 1, Test 4: 90

Student 2, Test 1: 78
Student 2, Test 2: 87
Student 2, Test 3: 78
Student 2, Test 4: 80

Student 3, Test 1: 99
Student 3, Test 2: 98
Student 3, Test 3: 97
Student 3, Test 4: 96

Student 1's average is 86.75.
Student 2's average is 80.75.
Student 3's average is 97.5.
```

(A Level)

Use a 2-D array to write a program that has 3 students and 4 test scores. Ask the user to enter the students' names and their test scores. Display their names and scores as follows:

```
This program will calculate the average of
three students' four tests' scores. Enter the students' names and their test scores:
Student 1's name :James
Test 1 is 78
Test 2 is 88
Test 3 is 87
Test 4 is 77

Student 2's name :Jason
Test 1 is 98
Test 2 is 78
Test 3 is 88
Test 4 is 89

Student 3's name :Drew
Test 1 is 99
Test 2 is 98
Test 3 is 88
Test 4 is 90

James's average is 82.5
Jason's average is 88.25
Drew's average is 93.75
```

Program Rubric

Programming Rubric

	Not Meet	Partly Meet	Meet Standard	Distinguished
Concepts	Don't demonstrate understanding of the concepts of each chapter.	Demonstrate some understanding of the concepts of each chapter.	Demonstrate a good understanding of the concepts of each chapter.	Demonstrate a thorough understanding of the concepts of each chapter.
Documentation	The code is not well documented without any comments.	The code is documented with a couple of comments at the beginning.	The code is fairly well documented with a few comments on a few lines.	The code is well documented with comments on every important line.
Output	The program doesn't runs well, some errors and output is incorrect. The format is not correct.	The program runs, two or three errors and output is not quite correct. The format is not quite correct.	The program runs well, one or two minor errors and output is correct but the format is not quite the same.	The program runs well, no errors and output is correct and the format is correct.
Design	The code is hard to read without any style. Each block is not easily found. The program is not easily debugged.	The code can be read with some effort. The program can be debugged with effort.	The code is fairly easy to read. With some style. Some blocks are easily found. The program is fairly easily debugged.	The code is easy to read with good and clear style. Each block is easily found. The program is easily debugged.