

Name: _____

Block: _____ Date: _____

Unit 8 Exam – Alternative Version

1. Consider the following method:

```
public static int mystery(int[] nums)
{
    int total = 0;
    for (int k = 0; k < nums.length / 3; k++)
    {
        total = total + nums[k];
    }
    return total;
}
```

Assume that the array test has been declared and initialized as follows.

```
int[] test = {1, 3, 2, 5, 8, 7, 0, 9, 2, 4};
```

What value will be returned as a result of the call `mystery(test)`?

- a. 4
- b. 6
- c. 11
- d. 13
- e. 26

2. Consider the following code segment.

```
int[][] matrix = new int[7][15];
```

Which of the following correctly gives the number of rows in the two-dimensional array matrix?

- a. `matrix[0]`
- b. `matrix.length`
- c. `matrix[matrix.length - 1]`
- d. `matrix[0].length`
- e. `matrix.length[0]`

3. Consider the following declaration for a two-dimensional array.

```
int[][] grid = new int[5][3];
int c = 0;
for (int i = 0; i < grid.length; i++)
{
    for (int j = 0; j < grid[i].length; j++)
    {
        grid[i][j] = c;
        c++;
    }
}
```

What element is displayed when the following line of code is executed?

```
System.out.println(grid[2][1]);
```

- a. 4
- b. 5
- c. 7
- d. 8
- e. 10

4. Consider the following method intended to swap the first and last rows in a two-dimensional array:

```
public static void swapRow(int[][] arr)
{
    /* missing code */
}
```

Which of the following correctly replaces `/* missing code */`?

- a.

```
for (int k = 0; k < arr[0].length; k++)
{
    int last = arr.length - 1;
    arr[0][k] = arr[last][k];
    arr[last][k] = arr[0][k];
}
```
- b.

```
for (int k = 0; k < arr[0].length; k++)
{
    int last = arr.length;
    arr[0][k] = arr[last][k];
    arr[last][k] = arr[0][k];
}
```
- c.

```
for (int k = 0; k < arr[0].length; k++)
{
    int last = arr.length - 1;
    int temp = arr[0][k];
    arr[0][k] = arr[last][k];
    arr[last][k] = temp;
}
```
- d.

```
for (int k = 0; k < arr[0].length; k++)
{
    int last = arr.length;
    int temp = arr[0][k];
    arr[0][k] = arr[last][k];
    arr[last][k] = temp;
}
```
- e. None of the above

5. Consider the following method declaration.

```
public static void increment(int[][] a)
{
    for (int r = 0; r < a.length; r++)
    {
        for (int c = 0; c < a[0].length; c++)
        {
            if (a[r][c] >= 0)
            {
                a[r][c]--;
            }
            else
            {
                a[r][c]++;
            }
        }
    }
}
```

Assume the 2D array, `matrix`, has been initialized to the following values:

```
4 6 -15
5 11 21
-11 -10 3
4 -10 -18
-21 14 -23
```

What is the value of `matrix` after the method call `increment(matrix)`?

a. 4 6 -15
5 11 21
-11 -10 3
4 -10 -18
-21 14 -23

b. 3 5 -15
4 10 20
-11 -10 2
3 -10 -18
-21 13 -23

c. 3 5 -16
4 10 20
-12 -11 2
3 -11 -19
-22 13 -24

d. 4 6 -14
5 11 21
-10 -9 3
4 -9 -17
-20 14 -22

e. 3 5 -14
4 10 20
-10 -9 2
3 -9 -17
-20 13 -22

6. The following code is meant to find the smallest value in an array.

```
double[][] list = /* Initialization not Shown */
double m = /* Initialization not Shown */
for (int i = 0; i < list.length; i++)
{
    for (int j = 0; j < list[i].length; j++)
    {
        if (list[i][j] < m)
        {
            m = list[i][j];
        }
    }
}
System.out.println(m);
```

What could m be set to in order for the code to work as intended?

- a. 100000
- b. 0
- c. -100000
- d. Double.MIN_VALUE
- e. Double.MAX_VALUE

7. Consider the following code.

```
int[][] matrix = new int[4][5];
```

Suppose we want to initialize matrix to the following rows and columns.

```
0 0 0 0 0
1 1 1 1 1
2 2 2 2 2
3 3 3 3 3
```

Which of the options below correctly initializes matrix?

I.

```
for (int i = 0; i < matrix.length; i++)
{
    for (int j = 0; j < matrix[i].length; j++)
    {
        matrix[i][j] = j;
    }
}
```

II.

```
for (int i = 0; i < matrix.length; i++)
{
    for (int j = 0; j < matrix[i].length; j++)
    {
        matrix[i][j] = i;
    }
}
```

III.

```
for (int i = 0; i < matrix.length; i++)
{
    for (int j = 0; j < matrix[i].length * 2; j += 2)
    {
        matrix[i][j] = j;
    }
}
```

- a. I only
- b. II only
- c. III only
- d. I and II only
- e. I, II and III

8. Consider the following code:

```
int[][] grid = /* code not shown */;
```

Which of the following could be used to calculate how many cells are in the array?

- a. `grid.length[0] * grid[0].length`
- b. `grid[0].length * grid.length`
- c. `grid[0].length`
- d. `grid.length`
- e. `grid.length * grid.length`

9. Consider the following method that is intended to return true if all the Strings in the ArrayList start with an uppercase letter:

```
public static boolean capitalized(String[][] a)
{
    /* Missing Code */
}
```

Which of the following could replace `/* Missing Code */` so that the method works as intended? (You may assume that all the Strings in 2-D array contain only letters.)

I.

```
for (String[] s : a)
{
    for (String st : s)
    {
        if (!st.toUpperCase().substring(0, 1).equals(st.substring(0, 1)))
        {
            return true;
        }
    }
}
return false;
```

II.

```
for (String[] s : a)
{
    for (String st : s)
    {
        if (!st.toUpperCase().substring(0, 1).equals(st.substring(0, 1)))
        {
            return false;
        }
    }
}
return true;
```

III.

```
for (String[] s : a)
{
    for (String st : s)
    {
        if (st.toUpperCase().substring(0, 1).equals(st.substring(0, 1)))
        {
            return false;
        }
    }
}
return true;
```

- a. I only
- b. II only
- c. III only
- d. I and III only
- e. II and III only

10. What does the following segment of code do?

```
int[][] a = /* initialization not shown */;
int sum = 0;
for (int i = 0; i < a.length; i++)
{
    for (int j = 0; j < a[0].length; j++)
    {
        if (i % 2 == 1)
        {
            sum += a[i][j];
        }
    }
}
```

- a. It finds the sum of every other element in the array.
- b. It finds the sum of the elements in the odd rows in the array.
- c. It finds the sum of the odd elements in the array.
- d. It finds the sum of all elements in the array.
- e. It finds the sum of the elements in the odd columns in the array.

11. Which option best describes what the following algorithm does?

```
int a [][] = /* initialization not shown */;
int j = 1;
for (int i = 0; i < a[0].length; i++)
{
    int temp = a[j + 1][i];
    a[j + 1][i] = a[j][i];
    a[j][i] = temp;
}
```

- a. Swaps columns 1 and 2.
- b. Swaps columns 2 and 3.
- c. Swaps rows 1 and 2
- d. Swaps rows 2 and 3.
- e. Initializes the values in the array.

12. You need a method to find the minimum value in every row of an array. Which of the following methods works as intended?

I.

```
public static int[] findMinList(int[][] a)
{
    int[] temp = new int[a.length];
    for (int i = 0; i < a.length; i++)
    {
        int min = a[i][0];
        for (int j = 0; j < a[0].length; j++)
        {
            if (a[i][j] < min)
            {
                min = a[0][j];
            }
        }
        temp[i] = min;
    }
    return temp;
}
```

II.

```
public static int[] findMinList(int[][] a)
{
    int[] temp = new int[a.length];
    for (int i = 0; i < a.length; i++)
    {
        int min = a[i][0];
        for (int j = 0; j < a[0].length; j++)
        {
            if (a[i][j] < min)
            {
                min = a[i][j];
            }
        }
        temp[i] = min;
    }
    return temp;
}
```

III.

```
public static int[] findMinList(int[][] a)
{
    int[] temp = new int[a.length];
    for (int i = 0; i < a.length; i++)
    {
        int min = a[i][0];
        for (int j = 0; j < a[0].length; j++)
        {
            if (a[i][j] < a[0][j])
            {
                min = a[i][j];
            }
        }
        temp[i] = min;
    }
    return temp;
}
```

- a. I only
- b. II only
- c. III only
- d. II and III only
- e. I, II and III

13. Consider the following code segment.

```
int[][] mat = new int[3][5];
for (int j = 0; j < mat.length; j++)
{
    for (int k = 0; k < mat[0].length; k++)
    {
        mat[j][k] = (k + j) * 2;
    }
}
```

What are the contents of `mat` after the code segment has been executed?

- a. $\{\{0, 2, 4\},$
 $\{2, 4, 6\},$
 $\{4, 6, 8\},$
 $\{6, 8, 10\},$
 $\{8, 10, 12\}\}$
- b. $\{\{2, 4, 6\},$
 $\{4, 6, 8\},$
 $\{6, 8, 10\},$
 $\{8, 10, 12\},$
 $\{10, 12, 14\}\}$
- c. $\{\{0, 2, 4, 6, 8\},$
 $\{0, 2, 4, 6, 8\},$
 $\{0, 2, 4, 6, 8\}\}$
- d. $\{\{0, 2, 4, 6, 8\},$
 $\{2, 4, 6, 8, 10\},$
 $\{4, 6, 8, 10, 12\}\}$
- e. $\{\{4, 6, 8, 10, 12\},$
 $\{6, 8, 10, 12, 14\},$
 $\{8, 10, 12, 14, 16\}\}$

14. Consider the following method

```
public static int[][] operation(int[][] mat, int c)
{
    int[][] result = new int[mat.length][mat[0].length];
    for (int j = 0; j < mat.length; j++)
    {
        for (int k = 0; k < mat[j].length; k++)
        {
            if (k >= c && j >= c)
            {
                result[j][k] = 0;
            }
            else
            {
                result[j][k] = mat[j][k];
            }
        }
    }
    return result;
}
```

The following code segment appears in another method in the same class.

```
int[][] m = {{1, 2, 4, 2},
             {3, 3, 5, 1},
             {2, 1, 3, 1},
             {1, 3, 2, 4}};

int[][] grid = operation(m, 2);
```

Which of the following represents the contents of grid as a result of executing the code segment?

- | | | |
|--|--|--|
| a. {{1, 2, 4, 2},
{3, 3, 5, 1},
{2, 1, 3, 1},
{1, 3, 2, 4}} | b. {{1, 2, 4, 2},
{3, 3, 5, 1},
{2, 1, 3, 1},
{1, 3, 2, 0}} | c. {{1, 2, 4, 2},
{3, 3, 5, 1},
{2, 1, 0, 0},
{1, 3, 0, 0}} |
| d. {{0, 0, 0, 0},
{0, 0, 0, 0},
{0, 0, 3, 1},
{0, 0, 2, 4}} | e. {{0, 0, 0, 0},
{0, 0, 0, 0},
{0, 0, 0, 0},
{0, 0, 0, 0}} | |

15. Consider the following definition

```
String[][] letters = {"c", "a", "t"},  
                     {"d", "o", "g"};
```

Which of the following code segments produces the output "tacgod"?

- a.

```
for (String l : letters)  
{  
    System.out.print(l);  
}
```
- b.

```
for (String[] row : letters)  
{  
    for (String l : row)  
    {  
        System.out.print(l);  
    }  
}
```
- c.

```
for (String[] row : letters)  
{  
    for (int k = row.length - 1; k >= 0; k--)  
    {  
        System.out.print(row[k]);  
    }  
}
```
- d.

```
for (int j = letters.length - 1; j >= 0; j--)  
{  
    for (int k = letters[j].length - 1; k >= 0; k--)  
    {  
        System.out.print(letters[j][k]);  
    }  
}
```
- e.

```
for (int j = 0; j < letters.length; j++)  
{  
    for (int k = 0; k < letters[j].length; k++)  
    {  
        System.out.print(letters[j][k]);  
    }  
}
```

16. Consider the following method.

```
public String mystery(String s1, String s2)
{
    String output = "";
    int len;

    if (s1.length() < s2.length())
    {
        len = s1.length();
    }
    else
    {
        len = s2.length();
    }

    for (int k = 0; k < len; k++)
    {
        output += s1.substring(k, k+1);
        output += s2.substring(k, k+1);
    }

    return output;
}
```

What is returned as a result of the call `mystery("Sally", "Joe")`?

- a. SJaole
- b. JSoael
- c. SJaolel
- d. SallyJoe
- e. Nothing is returned because an `IndexOutOfBoundsException` is thrown.

17. Consider the following code segment.

```
int[][] mat = new int[4][4];
int fill = 0;

for (int[] row : mat)
{
    for (int k = 0; k < row.length; k++)
    {
        row[k] = fill;
        fill++;
    }
}
System.out.println(mat[1][2]);
```

What is printed as a result of executing the code segment?

- a. 1
- b. 2
- c. 5
- d. 6
- e. 10

18. Consider the following method.

```
public static int operation(int[][] mat)
{
    int currentVal = mat[0][0];
    int result = 0;

    for (int j = 0; j < mat.length; j++)
    {
        for (int k = 0; k < mat[j].length; k++)
        {
            /* missing code */
        }
    }
    return result;
}
```

Which of the following should replace `/* missing code */` so that the method returns the index of the row which contains the largest value in the two-dimensional array?

- a.

```
if (mat[j][k] > currentVal)
{
    currentVal = mat[j][k];
    result = j;
}
```
- b.

```
if (mat[j][k] > currentVal)
{
    currentVal = mat[j][k];
    result = k;
}
```
- c.

```
if (mat[j][k] > currentVal)
{
    currentVal = j;
    result = mat[j][k];
}
```
- d.

```
if (mat[j][k] < currentVal)
{
    currentVal = mat[j][k];
    result = j;
}
```
- e.

```
if (mat[j][k] > currentVal)
{
    currentVal = k;
    result = mat[j][k];
}
```


19. Consider the following code segment.

```
int[][] mat = new int[4][6];
for (int r = 0; r < mat.length - 1; r++)
{
    for (int c = 0; c < mat[0].length - 1; c++)
    {
        if(c < 3 - r || c > 3 + r)
        {
            mat[r][c] = 1;
        }
        else
        {
            mat[r][c] = 0;
        }
    }
}
```

Which of the following represents mat after this code segment is executed?

a.

1	1	1	1	1	1
1	1	0	0	1	1
1	0	0	0	0	1
0	0	0	0	0	0

b.

1	1	1	1	1	1
1	1	1	0	1	1
1	1	0	0	0	1
0	0	0	0	0	0

c.

1	1	0	1	1	1
1	0	0	0	1	1
0	0	0	0	0	1
0	0	0	0	0	0

d.

1	1	1	0	1	1
1	1	0	0	0	1
1	0	0	0	0	0
0	0	0	0	0	0

e.

1	1	0	0	1	1
1	0	0	0	0	1
0	0	0	0	0	0
0	0	0	0	0	0

20. Consider the following method, which is intended to return an array which contains the minimum elements in each of the rows of a 2-dimensional array.

```
/** @param mat a 2-dimensional array
 * @return an array which contains the minimum elements of each row
 * in mat.
 */
public double[] minRows(double[][] mat)
{
    double[] mins = new double[mat.length];
    for (int k = 0; k < mat.length; k++)
    {
        double localMin = mat[k][0];
        for (double num : mat[k])
        {
            /* missing code */
        }
        mins[k] = localMin;
    }
    return mins;
}
```

Which of the following could be used to replace `/* missing code */` so that `minRows` will work as intended?

- | | | |
|--|---|--|
| a. <pre>if (num < localMin) { localMin = mat[k][num]; }</pre> | b. <pre>if (num < localMin) { localMin = num; }</pre> | c. <pre>if (mat[num] < localMin) { localMin = mat[num]; }</pre> |
| d. <pre>if (mat[k][num] < localMin) { localMin = num; }</pre> | e. <pre>if (mat[k][num] < mins[k]) { localMin = mat[k][num]; }</pre> | |