

Unit 8 Exam Practice

Started: Feb 29 at 10:35pm

Quiz Instructions

Question 1

1 pts

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)`?

☒ 6

☐ 4

☐ 26

☐ 13

☐ 11

Question 2

1 pts

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?

☒ matrix.length

☐ matrix[0]

☐ matrix.length[0]

☐ matrix[matrix.length - 1]

☐ matrix[0].length

Question 3

1 pts

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]);
```

☐ 8

☒ 7

- ☐ 5
- ☐ 4
- ☐ 10

Question 4**1 pts**

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 */`?

☐

```
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];
}
```

☐ None of the items listed.

☐

```
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;
}
```

☐

```
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];
}
```

☒

```
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;  
}
```

Question 5**1 pts**

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)?

- ☐ 3 5 -15
4 10 20
-11 -10 2
3 -10 -18
-21 13 -23
- ☐ 4 6 -15
5 11 21
-11 -10 3
4 -10 -18
-21 14 -23

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

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

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

Question 6

1 pts

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?

- ☐ -100000
- ☒ Double.MAX_VALUE
- ☐ 100000
- ☐ Double.MIN_VALUE
- ☐ 0

Question 7**1 pts**

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;
    }
}
```

☐ I, II and III

☐ III only

☐ I only

☒ II only☐ I and II only**Question 8****1 pts**

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?

☐ `grid.length * grid.length`☐ `grid[0].length`☐ `grid.length[0] * grid[0].length`☒ `grid[0].length * grid.length`☐ `grid.length`**Question 9****1 pts**

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;
```

☐ III only☐ II and III only☐ I and III only.☐ I only☒ II only**Question 10****1 pts**

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];
        }
    }
}
```

- ☐ It finds the sum of the odd elements in the array.
- ☐ It finds the sum of every other element in the array.
- ☐ It finds the sum of the elements in the odd columns in the array.
- ☒ It finds the sum of the elements in the odd rows in the array.
- ☐ It finds the sum of all elements in the array.

Question 11

1 pts

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;
}
```

- ☒ Swaps rows 2 and 3
- ☐ Swaps columns 1 and 2
- ☐ Swaps rows 1 and 2
- ☐ Swaps columns 2 and 3

- ☐ Initializes the values in the array.

Question 12

1 pts

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[i][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;  
}
```

☐ I, II and III only.

☐ II and III only

☒ II only

☐ I only

☐ III only

Question 13

1 pts

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?

☒

```
{{0, 2, 4, 6, 8},  
{2, 4, 6, 8, 10},  
{4, 6, 8, 10, 12}}
```

☐

```
{{0, 2, 4},  
{2, 4, 6},  
{4, 6, 8},  
{6, 8, 10},  
{8, 10, 12}}
```

☐

```
{{4, 6, 8, 10, 12},  
{6, 8, 10, 12, 14},  
{8, 10, 12, 14, 16}}
```

☐ {{2, 4, 6},
 {4, 6, 8},
 {6, 8, 10},
 {8, 10, 12},
 {10, 12, 14}}

☐ {{0, 2, 4, 6, 8},
 {0, 2, 4, 6, 8},
 {0, 2, 4, 6, 8}}

Question 14

1 pts

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?

☐ {{0, 0, 0, 0},
 {0, 0, 0, 0},

```
{0, 0, 3, 1},
{0, 0, 2, 4}}
```

☐

```
{{1, 2, 4, 2},
{3, 3, 5, 1},
{2, 1, 3, 1},
{1, 3, 2, 0}}
```

☒

```
{{1, 2, 4, 2},
{3, 3, 5, 1},
{2, 1, 0, 0},
{1, 3, 0, 0}}
```

☐

```
{{0, 0, 0, 0},
{0, 0, 0, 0},
{0, 0, 0, 0},
{0, 0, 0, 0}}
```

☐

```
{{1, 2, 4, 2},
{3, 3, 5, 1},
{2, 1, 3, 1},
{1, 3, 2, 4}}
```

Question 15

1 pts

Consider the following definition

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

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

☐

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

☒

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

```
{  
    System.out.print(row[k]);  
}  
}
```

☐

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

☐

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

☐

```
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]);  
    }  
}
```

Question 16

1 pts

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")`?

- ☐ Nothing is returned because an `IndexOutOfBoundsException` is thrown.
- ☐ JSoael
- ☒ SJaole
- ☐ SJaolel
- ☐ SallyJoe

Question 17

1 pts

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?

- ☐ 1
- ☐ 10
- ☐ 5
- ☐ 2
- ☒ 6

Question 18

1 pts

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?

☐

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

☐

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

☒

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

☐

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

☐

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



```

    result = j;
}

```

Question 19

1 pts

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?

☐

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

☐

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

☐

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

☐

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

☒

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

Question 20**1 pts**

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?

☒

```
if (num < localMin)
{
    localMin = num;
}
```

☐

```
if (mat[num] < localMin)
{
    localMin = mat[num];
}
```

☐

```
if (mat[k][num] < localMin)
{
    localMin = num;
}
```

☐

```
if (num < localMin)
{
    localMin = mat[k][num];
}
```

☐

```
if (mat[k][num] < mins[k])
{
    localMin = mat[k][num];
}
```

No new data to save. Last checked at 10:58pm

Submit Quiz