

Objective 7 - Java Static Two-Dimensional Arrays

33. Which of the following statements correctly declares a two-dimensional integer array?

- (A) **int Matrix[] = new int[5,4];**
- (B) **int Matrix[];**
Matrix = new int[5,4];
- (C) **int Matrix[][] = new int[5][4];**
- (D) **int Matrix[][];**
Matrix = new int[5][4];
- (E) Both choices C and D

34. What is the output of the program below?

```
public class Java1034
{
    public static void main(String args[ ])
    {
        int matrix[ ][ ];
        matrix = new int[3][4];
        for(int p = 0; p < 3; p++)
        {
            for(int q = 0; q < 4; q++)
                System.out.print(matrix[p][q] + " ");
            System.out.println();
        }
    }
}
```

(A) 0 0 0 0
0 0 0 0
0 0 0 0

(B) 0 0 0
0 0 0
0 0 0
0 0 0

(C) 0
0 0
0 0 0
0 0 0 0

(D) Compile Error

35. Consider the **mambo** object declaration below.

```
double mambo[ ][ ];
mambo = new double[4][5];
int r; // row index on mambo
int c; // column index of mambo
```

Which of the following statements stores the *column length* of **mambo**?

(A) **mambo.length**

(B) **mambo.rowLength**

(C) **mambo[r].length**

(D) **mambo[c].length**

36. What is the output of the program below?

```
public class Java1036
{
    public static void main(String args[ ])
    {
        int matrix[ ][ ];
        matrix = new int[3][4];
        for(int p = 0; p < 3; p++)
        {
            for(int q = 0; q < 4; q++)
                System.out.print(matrix[q][p] + " ");
            System.out.println();
        }
    }
}
```

(A) 0 0 0 0
0 0 0 0
0 0 0 0

(B) 0 0 0
0 0 0
0 0 0
0 0 0

(C) 0
0 0
0 0 0
0 0 0 0

(D) Error message

37. Consider the **mambo** object declaration below.

```
double mambo[ ][ ];
mambo = new double[4][5];
int r; // row index on mambo
int c; // column index of mambo
```

Which of the following statements stores the *row length* of **mambo**?

(A) **mambo.length**

(B) **mambo.rowLength**

(C) **mambo[r].length**

(D) **mambo[c].length**

Use the program below for questions 38-40. Each question will provide a different implementation of the **createSquare** method. The output shown is formatted for ease of reading. Technically, the columns will not line up so nicely.

```
public class Java3840
{
    public static void main (String args[])
    {
        int square[][] = new int[5][5];
        createSquare(square);
        displaySquare(square);
    }

    public static void createSquare(int[][] square)
    {
    }

    public static void displaySquare(int[][] square)
    {
        for (int r = 0; r < 5; r++)
        {
            for (int c = 0; c < 5; c++)
                System.out.print(square[r][c] + " ");
            System.out.println();
        }
    }
}
```

38. What will be the output of program **Java3840.java** with the **createSquare** implementation below?

```
public static void createSquare(int[ ][ ] square)
{
    int size = square.length;
    for (int r = 0; r < size; r++)
    {
        int q = 1;
        for (int c = r; c < size; c++)
        {
            square[r][c] = q;
            q++;
        }
    }
}
```

(A) 0 1 2 3 4
1 2 3 4 5
2 3 4 5 6
3 4 5 6 7
4 5 6 7 8

(B) 1 2 3 4 5
0 1 2 3 4
0 0 1 2 3
0 0 0 1 2
0 0 0 0 1

(C) 0 0 0 0 1
0 0 0 1 2
0 0 1 2 3
0 1 2 3 4
1 2 3 4 5

(D) 1 2 3 4 5
6 7 8 9 10
11 12 13 14 15
16 17 18 19 20
21 22 23 24 25

39. What will be the output of program **Java3840.java** with the **createSquare** implementation below?

```
public static void createSquare(int[ ][ ] square)
{
    int size = square.length;
    for (int r = 0; r < size; r++)
    {
        int q = r;
        for (int c = 0; c < size; c++)
        {
            square[r][c] = q + c;
        }
    }
}
```

(A) 0 1 2 3 4
1 2 3 4 5
2 3 4 5 6
3 4 5 6 7
4 5 6 7 8

(B) 1 2 3 4 5
0 1 2 3 4
0 0 1 2 3
0 0 0 1 2
0 0 0 0 1

(C) 0 0 0 0 1
0 0 0 1 2
0 0 1 2 3
0 1 2 3 4
1 2 3 4 5

(D) 1 2 3 4 5
6 7 8 9 10
11 12 13 14 15
16 17 18 19 20
21 22 23 24 25

40. What will be the output of program **Java3840.java** with the **createSquare** implementation below?

```
public static void createSquare(int[][] square)
{
    int size = square.length;
    int r = 0;
    int c = size / 2;
    square[r][c] = 1;
    for (int k = 2; k <= size*size; k++)
    {
        if (k % size == 1)
            r++;
        else
        {
            r--; c++;
        }
        if (r < 0) r = size-1;
        if (c == size) c = 0;
        square[r][c] = k;
    }
}
```

(A) 1 6 11 16 21
2 7 12 17 22
3 8 13 18 23
4 9 14 19 24
5 10 15 20 25

(B) 11 18 25 2 9
10 12 19 21 3
4 6 13 20 22
23 5 7 14 16
17 24 1 8 15

(C) 17 24 1 8 15
23 5 7 14 16
4 6 13 20 22
10 12 19 21 3
11 18 25 2 9

(D) 1 2 3 4 5
6 7 8 9 10
11 12 13 14 15
16 17 18 19 20
21 22 23 24 25

Objective 8 - Storing Objects in a Static Array

Use the class and program segment below for questions 41-42.

```
class Student
{
    private String name;
    private int age;

    public Student(String n, int a)
    {
        name = n;
        age = a;
    }

    public void showData()
    {
        System.out.println("Name: " + name);
        System.out.println("Age: " + age);
        System.out.println();
    }
}

Student students[] = new Student[numStudents];
for (int index = 0; index < numStudents; index++)
{
    System.out.print("Enter student's name ==> ");
    String name = stringInput.nextLine();
    System.out.print("Enter student's age ==> ");
    int age = intInput.nextInt();
    students[index] = new Student(name,age);
}
```

42. Which of the following describes the relationship of the objects?

- (A) The program uses a **Students** array object of **Student** objects.
- (B) The program uses a **Student** array object of **Students** objects.
- (C) The program uses a **Student** object of **Student** array objects.
- (D) The program uses a **Students** object of **Students** array objects.

Use the class and program segment below for questions 43-44.

```
class MyList
{
    private int[] intArray;
    private int size;

    public MyList(int s)
    {
        intArray = new int[size];
    }
}

MyList list = new MyList(10);
```

43. Describes the result of executing the statement **MyList list = new MyList(10);**

- (A) It instantiates a **MyList** object.
- (B) It instantiates an **int[]** object called intArray
- (C) It allocates spaces for ten **int** values.
- (D) All of the above.

44. Which of the following describes the relationship of the objects?

- (A) The program uses a **MyList** array object of **int** values.
- (B) The program uses a **MyList** object of **IntArray** values.
- (C) The program uses a **MyList** object with an **int[]** of **int** values.
- (D) The program uses a **IntArray** object of **MyList** values.

Objective 9 - Two-Dimensional Arrays and length

(This topic was also included in the earlier two-dimensional array section; this is a strict focus on length)

45. Consider the following two-dimensional array declaration.

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

Which of the following statements will assign the correct size to **rowSize**?

- (A) **int rowSize = matrix.length;**
- (B) **int rowSize = matrix[0].length;**
- (C) **int rowSize = matrix[1].length;**
- (D) **int rowSize = matrix[2].length;**
- (E) All of the above

46. Consider the following two-dimensional array declaration.

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

Which of the following statements will assign the correct size to **colSize**?

- (A) **int colSize = matrix.length;**
- (B) **int colSize = matrix[0].length;**
- (C) **int colSize = matrix[1].length;**
- (D) **int colSize = matrix[2].length;**
- (E) All of the above

47. Consider the following two-dimensional array declaration.

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

Which of the following statements will assign the correct size to **colSize**?

- (A) **int colSize = matrix[0].length;**
- (B) **int colSize = matrix[1].length;**
- (C) **int colSize = matrix[2].length;**
- (D) **int colSize = matrix[3].length;**
- (E) All of the above

48. What will be printed by the following program statement?

```
System.out.println(matrix[0].length);
```

- (A) The number of rows in a two-dimensional "square" static array.
- (B) The number of columns in a two-dimensional "non-ragged" array.
- (C) The number of columns in the top row of a two-dimensional static array.
- (D) The number of columns in the row with index[0] of a two-dimensional array.
- (E) All of the above

Objective 10 - Parameter Differences Between Simple Data Types and Arrays

49. Consider the following program.

```
public class Question49
{
    public static void main (String args[])
    {
        int p = 10;
        int q = 20;
        swap(p,q);
        System.out.println(p + " " + q);
    }

    public static void swap(int x, int y)
    {
        int temp = x;
        x = y;
        y = temp;
    }
}
```

What is printed as a result of executing the program?

- (A) 10 20
- (B) 20 10
- (C) 10 10
- (D) 20 20
- (E) 0 0

50. Consider the following program.

```
public class Question50
{
    public static void main (String args[])
    {
        int[] list = {1,2,3,4,5,6,7,8,9};
        swap(list,3,4);
        System.out.println(list[3] + " " + list[4]);
    }

    public static void swap(int[] x, int p, int q)
    {
        int temp = x[p];
        x[p] = x[q];
        x[q] = temp;
    }
}
```

What is printed as a result of executing the program?

- (A) 3 4
- (B) 4 3
- (C) 4 5
- (D) 5 4
- (E) ArrayIndexOutOfBoundsException

