**Java**

**Array Review**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Questions 1 - 6 all refer to the same array.**

Scanner sc = new Scanner(System.in);

**1)** Declare and initialize an array of 20 doubles that are all 0.

double[] pranav = new double[20];

**2)** Write a for loop that sets each double in the array to twice its index. The first three elements should be: 0, 2, 4

for(int i = 0; i < 20; i++){

int l = 2 \* i;

pranav[i] = l;

}

**3)** Print the array on one line using this format (your values will be different): [0, 0, 0, 0]

System.out.print('[');

for(int i = 0; i < 20; i++){

System.out.print(pranav[i]);

if(i < 19) System.out.print(", ");

}

System.out.println(']');

**4)** Explain the difference between a value in an array, and an index.

value can change, the index is predetermined

index is the spot number, the value is what is saved at that spot

**5)** What is the first index in the array you made? Is that always the first index?

0, always first

**6)** What is the last index in the array you made? Is that always the last index?

19, no: length - 1

**7)** The following code segments are meant to create an array with 10 ints, and then have the user type in a value for each int. Some of them do not work (uh oh). If there is something wrong, explain what it is and fix it with the least amount of changes possible.

|  |  |  |  |
| --- | --- | --- | --- |
| **a)** | int[] numbers = new int[10];  for(int i = 1-1; i <= 10-1; i++){  //System.out.print(number);  int number = sc.nextInt();  numbers[i] = number;  }  number is not a thing, no input | **b)** | int[] numbers = new int[9+1];  for(int i = 0; i < 10; i ++){  numbers[i] = sc.nextInt();  System.out.print(numbers[i]);  } |
| **c)** | int[] numbers={0,0,0,0,0,0,0,0,0,0};  for(int i = 0; i < 9+1; i++){  int input=sc.nextInt();  //System.out.print(input);  numbers[i] = input;  } | **d)** | int[] numbers={0,0,0,0,0,0,0,0,0,0};  int index = 0;  while(index <= 9){  //System.out.print(index[numbers]);  numbers[index] = sc.nextInt();  index++;  } |

**8)** Write a method that returns the sum of an array of ints.

public int sum(int[] arr){

int sum = 0;

for(int i = 0; i < arr.length; i++){

sum += arr[i];

}

return sum;

}

**9)** What is the difference between a one dimensional array, and a two dimensional array?

one dimensional arrays hold one value for each index

two dimensional arrays hold an array of values for each index

**10)** Is there really a data type in java called a "two dimensional array?" Explain.

no, it is an array of arrays

**11)** What is the difference between printing an array in row major order, compared to column major order? Is one way more correct?

sideways

row major: [row][column]

col major: [column][row]

neither is more correct - however you want to look at it

**12)** Complete the method that prints the array matrix in row major order.

public static void printRowMaj(int[][] matrix){

for(int i = 0; i < matrix.length ;i++ ){

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

System.out.print(matrix[i][j] + "\t");

}

System.out.println();

}

**13)** Complete the method that prints the array matrix in column major order.

public static void printColMaj(int[][] matrix){

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

for(int j = 0; j < matrix.length ; j++){

System.out.print(matrix[j][i] + "\t");

}

System.out.println();

}

**14)** Write a method that takes a one dimensional array as a parameter, and copies those values into every array inside of a two dimensional array. For example:

int[] oneDValues = {2, 4, 6, 8};

int[][] twoDValues = new int[5][4];

copyOneToTwoD(oneDValues, twoDValues);

printRowMaj(twoDValues); // prints the array in row major order

//Should print:

2 4 6 8

2 4 6 8

2 4 6 8

2 4 6 8

2 4 6 8

public static void copyOneToTwoD(int[] oneDValues, int[][] twoDValues){

for(int r = 0; r < twoDValues.length; r++){

for(int c = 0; c < twoDValues[0].length; c++){

twoDValues[r][c] = oneDValues[c];

}

}

}

**15)** Complete the method that returns a one dimensional array containing the first n even integers in the two dimensional array matrix, reading the array in column major order starting from the top left.

For example:

matrix: (in column major order)

1 2 3 4

5 6 7 8

9 10 11 12

getEvens(matrix, 3) should return:

[2, 4, 6]

public int[] getEvens(int[][] matrix, int n){

int[] result = new int[n];

int index = 0;

for(int r = 0; r < matrix[0].length && index < n; r++){

for(int c = 0; c < matrix.length && index < n; c++){

if(matrix[c][r] % 2 == 0){

result[index] = matrix[c][r];

index++;

}

}

}

return result;

}