**YouTube tutorial 33 – Multidimensional arrays**

The code used is as follows:

**class** apples{

**public** **static** **void** main(String [] args){

**int** firstarray[][]={{3,4,5,6,7},{1,2}};

**int** secondarray[][]={{22,23,24,25},{30,31,32},{64}};

}

}

When we want to have two or more rows, we write down double square brackets. This way, we’re letting the computer know we will be inputting 1+ list of indexes.

Example for calling indexes:

firstarray[0] [0] is 3.

firstarray[1] [1] is 2.

secondarray[2] [0] is 64.

**YouTube tutorial 34 – Table for multi arrays**

**class** apples{

**public** **static** **void** main(String[] args){

**int** firstarray[][]={{8,9,10,11},{12,13,14,15}};

**int** secondarray[][]={{30,31,32,33},{43},{4,5,6}};

System.*out*.println("This is the first array");

*display*(firstarray);

System.*out*.println("This is the second array");

*display*(secondarray);

}

**public** **static** **void** display(**int** x[][]){

**for** (**int** row=0;row<x.length;row++){

**for** (**int** column=0;column<x[row].length;column++){

System.*out*.print(x[row][column]+"\t");

}

System.*out*.println();

}

}

}

This is pretty much self-explanatory, followed with the comment help.

**Notable comment:**

Maybe people are failing to understand this easily because Bucky hasn't told us something which might seem obvious but people might not realize.

If u have a 2D Array, int arry[][] = { {1,2,3}, {4,5,6} }; arry.length = 2. Why? Because in the main array "Arry" There are 2 smaller arrays. That's why it will give you the number of arrays inside your array.

If you want to know the length of one of the smaller arrays inside Arry then you would have to do: arry[0].length to know the length of the first smaller array (row). Why? Because 0 means the first as speaking in computer language.

arry[1].length would give u the length of the second smaller array or second row.﻿