**YouTube tutorial 22 – Methods in arrays**

The code used in this video is as follows:

**class** apples{

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

**int** bucky[]={3,4,5,6,7};

*change*(bucky);

**for**(**int** y:bucky)

System.*out*.println(y);

}

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

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

x[counter]+=5;

}

}

**Result:**

8

9

10

11

12

**Notable comment regarding the code:**

I wish bucky would talk about the interesting fact that arrays are passed by reference. For anyone not familiar with passing by reference consider the following code:

public static void main(String args[]){

int bucky = 5;

change(bucky);

System.out.println(bucky);

}

public static void change(int x){

x += 3;

}

What does this print out? No it's not 8. It's 5! So why does the change method in this video change the values of the array but my code leaves bucky unchanged. It's because arrays are passed by reference and integers are not. Basically what is happening is when you use an array as parameter to a method you are passing the location of the array on the hard drive(not the values in the array) into the method. And so when you make changes in the method you are changing what is stored at the original place on the hard drive and hence you are changing the original array. This does not happen with integers. In my code above we are passing a copy of the value of bucky(i.e., 5) to the change method and that copy is what is changes not the original. To make my program do the change to bucky we must do something like:

public static void main(String args[]){

int bucky = 5;

bucky = change(bucky);  // I changed this line

System.out.println(bucky);

}

public static int change(int x){  // I changed this line

x += 3;

return x;       // I added this line

}

While this is really a more important topic for when studying C and C++, I still think it is at least important to know that arrays are handled differently than some other data types when passing them into a method.﻿

**Important notes:**

* In other words, unlike integers, arrays have its values changed instantly instead of making copies like integers.
* Integers are passed as copies in methods.