7.2 Using Arrays

Example: Store the numbers 0 - 1000 in an array:

```
int[] numbers = new int[1001];
```

This creates the array and initializes all value to 0.

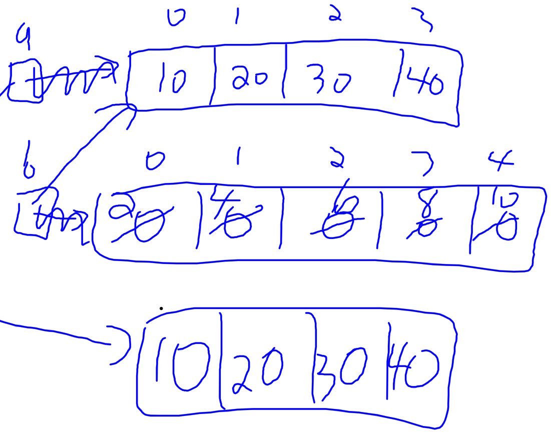
```
for (int i = 1; i < numbers.length; i++)
   numbers[i] = i;
```

We'd normally start the loop at 0, but `numbers[0]` is already 0.

We could do `numbers[1] = 1; numbers[2] =2;` and so on but it would be too tedious. Instead we use a for loop - this is why arrays are so useful.

If you try to access `numbers[1001]` you will get an ArrayIndexOutOfBoundsException and the program will crash. Although there are 1001 elements, 1000 is the biggest valid index value.

```
int[] a = {10, 20, 30, 40};
int[] b = new int[5];
for (int i = 0; i < b.length; i++)
   b[i] = (i + 1) * 2;
println(a == b);  —> false
                  they point to different arrays
b = a;
println(a == b);  —> true
                  they point to the same array
a = new int[4];
for (int i = 0; i < a.length; i++)
   a[i] = (i + 1) * 10;
println(a == b);  —> false
```



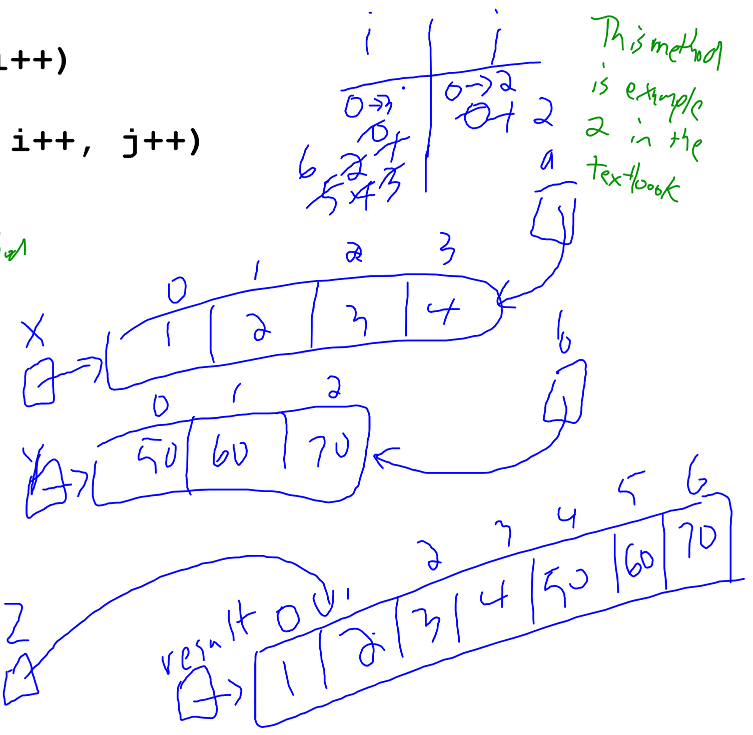
```
float[] join(float[] a, float[] b)
{
    float[] result = new float[a.length + b.length];
    int i,j;
    for(i=0; i<a.length; i++)
        result[i] = a[i];
    for (j=0; j<b.length; i++, j++)
        result[i] = b[j];
    return result;
}
```

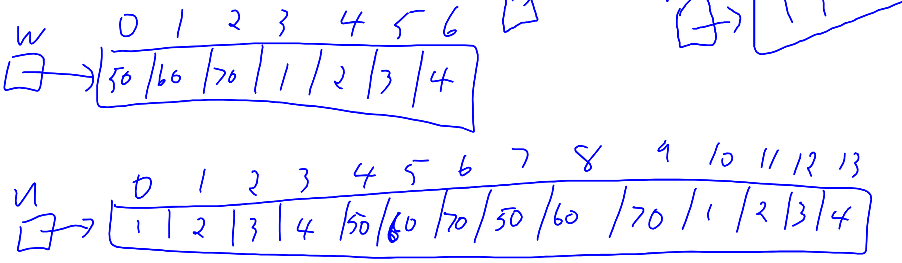
You can pass arrays as parameters and return arrays from a method

This method is example 2 in the textbook

Here is an example of using this method

```
int x = {1,2,3,4}
int y = {50,60,70}
int z = join(x,y);
int w = join(y,x);
int u = join(z,w);
```





Read section 7.2 in the textbook.

Do Exercise 7.2 #1-5

I will post solutions to #1, 3 - 5 on Thursday.

I would like you to email me (gerry.heffernan@tdsb.on.ca) your answers to #2 by Tuesday, April 14 (either type into email, or attach a word document or picture of handwritten work)