**AP Computer Science – Arrays**

**Array Vocabulary**

*array* - A data structure that allows you to create several different “versions” of the same variable. They use a single name with different index values to keep them separated.

index - A number that specifies the location in the array that you wish to examine.

element - This terms is used to refer to a single value stored at a single index of an array. In other words, an array containing 25 elements stores 25 different values.

traverse - To examine all elements in an array.

search - To look for element(s) in an array that match a certain criteria.

sort - To change the order of the elements in an array based on a certain criteria.

**Array Indices**

Each number in the square brackets is called an index. As long as we change the index value, we are reference a different element in the array.

**Declaring an Array**

An array can hold any type of data we’ve discussed so far. [boolean, int, double, char, String, etc..] You basically are declaring how many different “versions” of the variable you are going to permit.

**Method # 1 - Declaring an empty array of a given size.**

**int [ ] nums = new int [10];**

This creates an array with ten elements that would be referenced using num[0] to num[9].

**Method # 2 - Declaring an array and filling it in with values**

**int [ ] nums = { 4, 5, -2, 22, 301, -23, 0 };** This creates an integer array with seven values

**Putting Data into an Array**

Students[0] = “Rush Ayad”;

Students[1] = “Andy Truong”;

Students[2] = “Phuong Bui”;

Etc…

An array can function something like a database, only it only has one kind of information, in this case, the names of the students.

**Array Length**

The size of an array is found through *.length*

Ex: For the nums array above (Method # 2), if we did *System.out.println(nums.length);* 7 would be the output.

**Array Assignment**

**Declaring Arrays**

Declare a boolean array called *achievements* with 20 elements.

Declare an array called *scores* that will hold 35 double values.

Declare an integer array called *primes* that stores the first 5 prime numbers, starting at 2. (Declare and populate)

Declare a String array called names that stores all of the first names of your immediate family. (Declare and populate)

\*\*For the next section, assume we have an array of 20 integers called *nums*.

**Searching an Array for a specific value**

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

{

 if(nums[x]==10) System.out.println("10 was found at index " + x);

}

**Sum all Elements in an Array**

int sum = 0;

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

{

 sum += nums[x];

}

**Output All Elements in an Array**

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

{

 System.out.println(nums[x]);

}

**Adding 1 to Every Element in an Array**

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

{

 nums[x]++;

}

**Finding the Index of a Value in an Array**

int z = 22;

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

{

 if (nums[x]==z) System.out.println(z + “ was found at index “ + x);

}

**Sending Arrays into Methods**

Arrays are handled differently than regular variable. They are referenced by their location in memory. For this reason, when we send an array into a method, if the array is changed inside the method, it will be changed outside the method, even if the array isn’t returned.

Public static void main(String [] args){

int [ ] nums = {1, 2, 3, 4};

mult(nums);

}

public static void mult(int[] n){

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

 n[x] \*= 2;

}

If I were to output the contents of the nums array in the main method, it would be {2, 4, 6, 8}