Array Data Structure

Introduction

An array is an aggregate data structure that is designed to store a group of objects of the same or different types. Arrays can hold primitives as well as references. The array is the most efficient data structure for storing and accessing a sequence of objects.

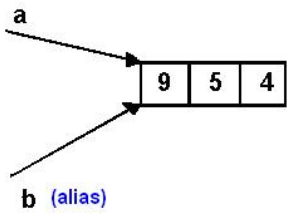
Here is the list of most important array features you must know (i.e. be able to program)

- copying and cloning
- insertion and deletion
- searching and sorting

You already know that the Java language has only two data types, primitives and references. Which one is an array? Is it primitive? An array is not a primitive data type - it has a field (and only one), called *length*. Formally speaking, an array is a reference type, though you cannot find such a class in the Java APIs. Therefore, you deal with arrays as you deal with references. One of the major diffeences between refeences and primituives is that you cannot copy arrays by assigning one to another:

```
int[] a = {9, 5, 4};
int[] b = a;
```

The assignment operator creates an alias to the object, like in the picture below



Since these two references **a** and **b** refer to the same object, comparing them with the double equal sign "==" will always return true. In the next code example,

```
int [] a = {1,2,3};
int [] b = {1,2,3};
```

a and **b** refer to two different objects (though with identical contents). Comparing them with the double equal sign will return false. How would you compare two objects with identical contents? In short, using the `equals` method. For array comparison, the Java APIs provides the *Arrays* class.

Go to this website and READ THOROUGHLY:

https://www.cs.cmu.edu/~adamchik/15-121/lectures/Arrays/arrays.html