

† Programming  
Assignment †@  
Java Collections

---

***PART I: Arrays***

1. Create an object that has a single member variable: an array of 10 integers. Use the object's constructor to initialize the array with a fixed set of numbers (you decide which ones) in random order.
2. Now add two methods to the object,
  - a. `getArray()` which returns the original array, and
  - b. `getSortedArray()` which returns a sorted version of the array.
  - c. Finally add a main method which will create an instance of the object, and print out the results of calling each method.
3. Extend the main method so that it makes a copy of the first five elements in the sorted array, using the `System.arraycopy` method. Print out the new array to confirm that the copy has happened correctly.

## ***PART II: Lists***

***Programming Tip:*** The *instanceof* operator can be used to test the type of an object. E.g. `object instanceof Person` will be true, if the object is an instance of the class, or implements that interface.

1. Write a class that holds a List of Person objects, (use the `ap.person` package found on the course website).
  - a. In the constructor initialize its member variable to use an `ArrayList`, and then
  - b. Populate it with some `Student`, `Lecturer` and `Employee` objects
2. Next, implement a method to walk through the list using a for loop, and print out the details of each Person held in the list
3. Implement a second method that iterates through the list using an `Iterator`. Have the method count up how many instances of the different type of objects (i.e. `Students`, `Lecturers`, etc) there are and print it out.
4. Add another method that also walks through the list, but instead puts each type of object in a separate list (i.e. a list for `Student` objects, a list for `Employee` objects, etc). Then have it print out the contents of each of the new lists.
5. Add another method that, similarly to the above, walks through the list of `Person` objects and builds up a new list of `Employee` objects. Then use the `removeAll` method to remove those employees from the original list. Print the list afterwards to confirm they have been removed.

## **PART I @Maps**

### **Q 3.1**

Get Strings from the command line, present in the console a vertical bar chart of the frequency of each letter in the input.

- Treat small and capital letters the same -- as capital
- Ignore any char that is not an English letter

#### **Example**

For the following input:

Hey how are you?

We expect the following chart:

<b>A</b>	<b>1</b>
<b>E</b>	<b>2</b>
<b>H</b>	<b>2</b>
<b>O</b>	<b>2</b>
<b>R</b>	<b>1</b>
<b>Y</b>	<b>2</b>
<b>U</b>	<b>1</b>
<b>W</b>	<b>1</b>

### **Q 3.2**

1. Create a crude Person database by writing an object that holds a Map of Person objects (use the ap.person package found on the course website).
  - a. Populate this map with a fixed list of people in the constructor of the object.
  - b. Place each Person object into the Map using their email address.
  - c. Add a method to the object that accepts an email address as a parameter, and returns the Person from the “database”
2. Write a main method that queries the database for several email addresses, printing out the details of each Person when found. Make sure you handle cases where the person isn't in the database

3. Add methods to print out the contents of your database by iterating through the contents of the Map