University of the Western Cape | Computer Science Department

CSC211: Data Structures I | Assignment 1 | Due: First Week of Term 2, 2020

A. Basic I/O: Your objectives are as follows:

1. Read the csv file containing the requisite data. This should be done in a method called "READ_FILE()".
2. Parse the content of the csv file and store into an appropriate data structure object. This should also be included in the method READ_FILE() created above. The String split method might be useful.
3. Write a code to return the number of items stored in the object. This should be in a method called "SIZE()".
4. Within a method named "ACCEPT_INPUT()", accept an input from a user and adds it to the data structure object.

B. Search: Task of finding and/or deleting.

1. Write a method called "SEARCH()", which accepts either a name or age as input from a user and check if its present in the object list. If found, your method should output: The Name(s), Age(s), Level(s) & Nationality(ies) of the requested input. If not found it should simply output "Not Found"
2. Similar to B1 above, write a method called "DELETE()", which accepts a name from a user, calls the SEARCH method above and if found, removes the item from the data structure object.

C. Sorting: Your next task is to sort the content of the object using Bubble sort.

1. Create two methods named: ITERATIVE_BUBBLE() and RECURSIVE_BUBBLE()
2. Create a timestamp as such START_TIME = System.nanoTime()
3. Call the ITERATIVE_BUBBLE method and pass the data structure object you created in A to it as a parameter.
4. Write a code to perform an **iterative** bubble sort on only the Age attribute
5. Create a timestamp END_TIME, right after the bubble sort is done.
6. Display the sorted structure
7. Repeat steps 2 to 6 for the RECURSIVE_BUBBLE sort, however in this case you should perform a **recursive** bubble sort on the Age attribute.

D. Report

1. Your report should be formatted as follows:
    a. Add your Full name and Student number as header for every page
    b. Full source code
2. In **one (1) or two (2) sentences**, state what your observation(s) were with respect to the time spent running the iterative versus recursive bubble sort.
3. Comment on your observation regarding the runtime speed. Explain the possible reason(s) why one is faster/slower than the other or why they are of equal speed (if that is your observation). **This should be in about five (5) sentences or less**.

Notes & Hints:

1. Please **strictly adhere** to the naming convention and method names given – they are given for a reason.
2. Program should be written in Java

3. Minimize the use of in-built methods as the goal of this assignment is to test your understanding of these concepts.
4. For the data structures:
    a. You might want to consider making a private class to hold the name, age, level and nationality of each person.
    b. You might also consider using an ArrayList to store the list of persons. That is put the list of person objects in an ArrayList.
5. Skeleton codes are provided below. Fill in the missing entries and implement other necessary methods.
6. Finally and in line with the University's regulations, plagiarism is a serious offence; it is not taken lightly by the department and the University as a whole. It is therefore in your best interest not to copy. Doing so would not only fetch you a mark of zero but might result in further disciplinary actions taken against you.

```
public class assignment1 {

//////////////Task A: Basic I/O////////////////////////////////////////////
private class person{
public person(String…) {
//constructor - use to initialize the attributes of each person
}
@Override
public String toString() {
return "person [name=" + name + ", age=" + age + ", level=" + level + ",
nationality=" + nationality + "]";
}
}
private ArrayList<person> people;/*ArrayList datastructure to hold list of people*/
public int SIZE() {        }
public ArrayList<person> read_file() {         …
BufferedReader buff = null;
try {
buff = new BufferedReader (new FileReader("Assignment1.csv"));
       …
}
} catch (IOException e) {e.printStackTrace();}
return…
}
public void ACCEPT_INPUT() {
Scanner sc = …
System.out.println("Enter details for a new person in ONE line. \nSeparate Name,
Age, Level and Nationality with a COMMA (,) ");
….
}
//////////////////////////////////////////TASK 2: FIND & DELETE
//////////////////////////////
public person SEARCH() {
person p = null;
Scanner sc …
System.out.println("Enter the Name or Age of Person being sought after ");
String r = sc.nextLine();
if (Character.isDigit(r.charAt(0))) { //test for Age
//search based on Age
…
}
}
else { //searching based on Name
…
}
}
return …;            }
public void DELETE() {
person x = search();
if (x != null) {
…
}
}
//////////////////////////////////TASK 3: BUBBLE SORT/////////////////////////////
```