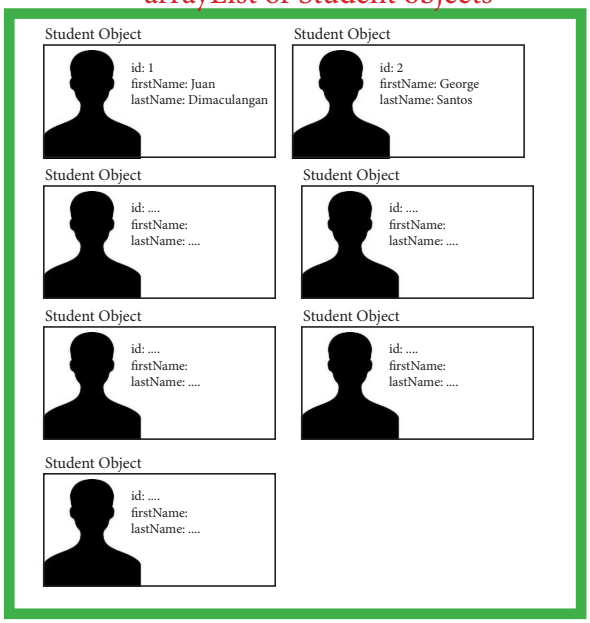
PART 4
JAVA OOP TUTORIAL - I/O
By CodyScott for PHCorner.net

Like I said, guys, this part 4 should be very very easy.
SORTING
- Basically it sorts this list of student objects.
- It takes 2 student object and compare by first name, or last name, or ID.

arrayList of Student objects



- Just add another class to our system and call it Sort.

Sort class

```
1  import java.util.Comparator;
2
3  public class Sorter {
4
5      public static final Comparator<Student> FIRSTNAME_ASC = new Comparator<Student>() {
6          public int compare(Student student1, Student student2) {
7              return student1.getFirstName().compareTo(student2.getFirstName());
8          }
9      };
10
11     public static final Comparator<Student> FIRSTNAME_DESC = new Comparator<Student>() {
12         public int compare(Student student1, Student student2) {
13             return student2.getFirstName().compareTo(student1.getFirstName());
14         }
15     };
16
17 }
18
```

Code shows sorting ASCENDING or DESCENDING using firstName.
Can you guys figure out kung paano kung lastName or ID ang iso-sort? (Kayang kaya nyo na yan dagdagan)

- And test it and change the MainApp slightly. Ganito:

```java
public class MainApp {
    static final String SOURCE_FILE = "students.txt";
    static List<String> studentRecords;
    static List<Student> studentsList;


    public static void main(String[] args) throws IOException {
        //1. initialize our MainApp
        initialize();

        //2. Sort First Name descending
        Collections.sort(studentsList, Sorter.FIRSTNAME_DESC);

        //2. Display total number of students
        displayNumberOfStudents();

        //3. Display details of students
        displayAllStudentsInformation();
    }


    //helper method
    private static void initialize() throws IOException{
        //we create an instance of StudentReader
        StudentReader tagabasa = new StudentReader();
        //we perform the method to read the txt file and store it in records array
        studentRecords = tagabasa.readPerRecord(SOURCE_FILE);
        //then we perform another method to read the records array, split it and create student object
        //then store it in students array
        studentsList = tagabasa.readAndCreateStudentList(studentRecords);
    }
    //helper method
    private static void displayNumberOfStudents() {
        System.out.format("There are %s total students%n%n", studentsList.size());
    }
    //helper method
    private static void displayAllStudentsInformation() {
        System.out.println("They are:");
        for(Student aStudent : studentsList){
            System.out.format("%s, %s - SN#%s%n", aStudent.getLastName(), aStudent.getFirstName(), aStudent.getID());
        }
    }

}
```

since declared as STATIC yung method under Sort Class, hindi na kailangan gumawa ng instance of Sort class bago gamitin ang method. THE ONLY CHANGE DO it before displaying number of students and students information

OK, hanggang sa muli. You should have a fully working program that can read, sort and display students' information. SORT does all the sorting. You can add more code to sort lastName and ID.

Next Lesson, siguro how to write TO A FILE naman.

Please do not forget to HIT LIKE or reply a message.
Salamat.