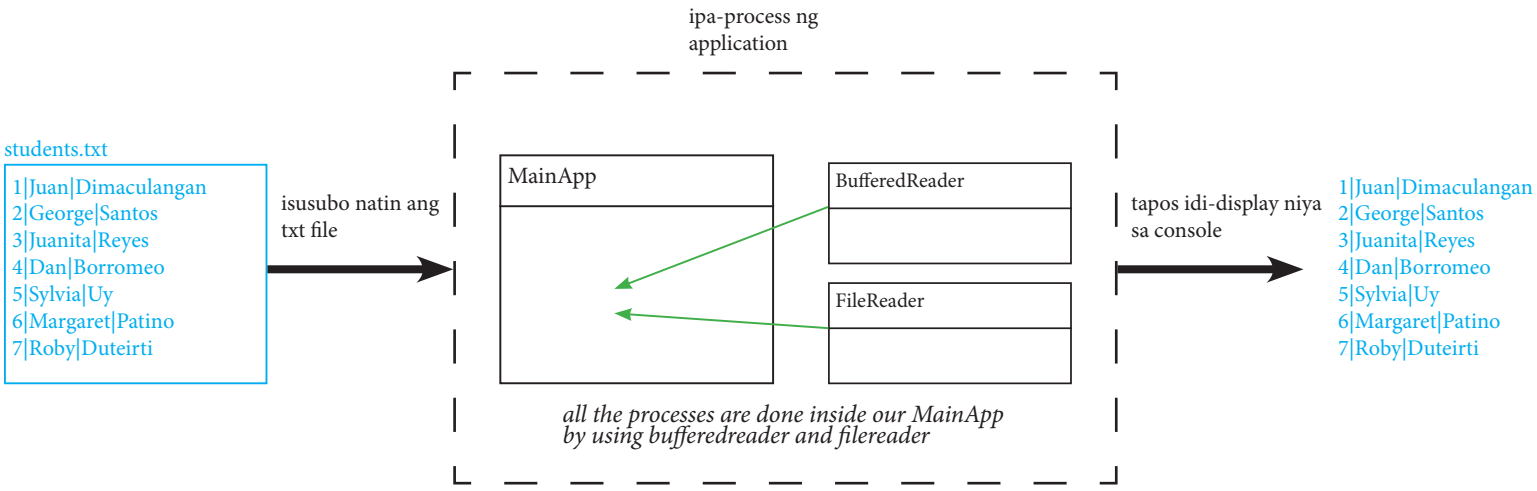
PART 2
JAVA OOP TUTORIAL - I/O
By CodyScott for PHCorner.net

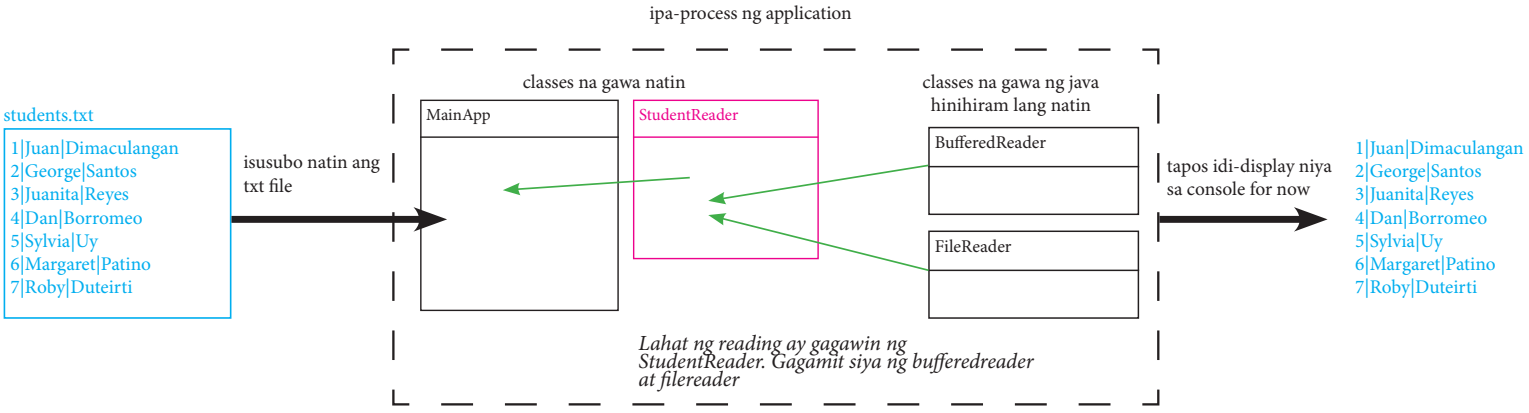
Currently our application has this, visually:



Since gusto natin maging Good OOP Programmer. At saka gusto natin i-practice and OOP, let's slightly modify our application. Gusto natin as much as possible, malinis at organize ang ating MainApp.
We want to assign the READING part to a new Class.

Let's call it StudentReader.

Here's the new visual.



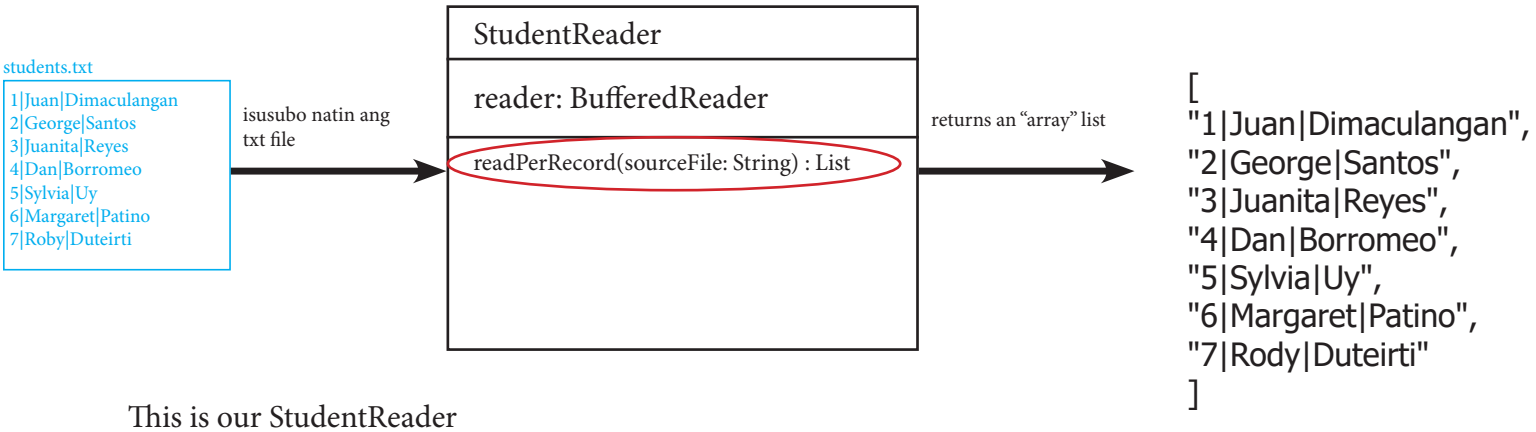
This is our new MainApp

```java
import java.io.IOException;
import java.util.List;

public class MainApp {
    static final String SOURCE_FILE = "students.txt";

    public static void main(String[] args) throws IOException {
        //1. reader the txt file
        readTheTxtFilePerRecord();
    }

    //helper method, para lang hindi maging cluttered yung main method natin
    private static void readTheTxtFilePerRecord() throws IOException {
        //we instantiate an OBJECT of our class StudentReader
        StudentReader tagabasa = new StudentReader();

        //we catch the returned value na arraylist
        List<String> mgaRecords = tagabasa.readPerRecord(SOURCE_FILE);

        //temporarily, display lang natin ang laman ng returned arraylist
        for(String record : mgaRecords){
            System.out.println("Record #-" + record);
        }
    }

}
```

take note: malinis ang ating main method

we used a helper method to do the "dirty work"

Let's visualize ang ating StudentReader



This is our StudentReader

```java
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;
import java.util.ArrayList;
import java.util.List;

public class StudentReader {
    BufferedReader reader;

    public List<String> readPerRecord(String sourceFile) throws IOException{
        //1. We instantiate an OBJECT from class ArrayList
        //to hold our records (or per line) into an array
        //magi ganito ang laman niya
        //["1|Juan|Dimaculangan","2|George|Santos","3|Juanita|Reyes","4|Dan|Borromeo","5|Sylvia|Uy","6|Margaret|Patino","7|Rody|Duteirti"]
        List<String> recordsList = new ArrayList<String>();

        //2. now instantiate an OBJECT reader from BufferedReader class
        reader = new BufferedReader(new FileReader(sourceFile));

        //3. let's declare a String para mag store ng per line data
        String aRecord;

        //4.huhugutin na natin line per line from the txt file to out aRecord string
        while((aRecord = reader.readLine())!=null){
            //5. then store that line sa ating recordsList array
            recordsList.add(aRecord);
        }
        //6. now we return this array sa mga parts ng program natin na may gusto
        return recordsList;
    }



}
```

Our revised MainApp

```java
import java.io.IOException;
import java.util.List;

public class MainApp {
    static final String SOURCE_FILE = "students.txt";

    public static void main(String[] args) throws IOException {
        //1. reader the txt file
        readTheTxtFilePerRecord();
    }

    //helper method, para lang hindi maging cluttered yung main method natin
    private static void readTheTxtFilePerRecord() throws IOException {
        //we instantiate an OBJECT of our class StudentReader
        StudentReader tagabasa = new StudentReader();

        //we catch the returned value na arraylist
        List<String> mgaRecords = tagabasa.readPerRecord(SOURCE_FILE);

        //temporarily, display lang natin ang laman ng returned arraylist
        for(String record : mgaRecords){
            System.out.println("Record #-" + record);
        }
    }

}
```

malinis na main method. only 1 line of code.

we instantiated an object from StudentReader

now we can use the method inside StudentReader ...tapos sinalo natin yung ire-return niyang value

tapos idi-display lang natin yung laman to make sure the gumagana ang ating app.

Lessons to learn from this part 2
1) Readable at organized na MainApp
2) Divide tasks into smaller and manageable component (hence our StudentReader)
3) Make use of Java available classes
4) If you can visualize it, you can code it

OK, guys, hanggang dito muna.
PLEASE TRY TO Follow, digest, understand, practice practice practice.
This is your only ticket to pass your Java course..... or become a good Java programmer.

Hanggang sa muli. Please hit like or reply to my post.