```java
package fileIO;

import java.io.File;

// using an ArrayList collect many students
public class StudentList {
    private ArrayList<Student> studentList;

//create an empty arraylist
    public StudentList() {
        studentList = new ArrayList<Student>();
    }

//add a student s to the collection
    public void add(Student s) {
        studentList.add(s);
    }

//returns a report with one line per person
// traverses the array list,
//getting one element at a time
    public String getAllStudents() {
        String report = "";
        for (Student s : studentList) {
            report += s.getName().getFullName() + "\n";
        }
        return report;
    }

//returns the number of elements in the list
    public int getSize() {
        return studentList.size();
    }

//returns the Student object at specified index position
    public Student getAtIndex(int index) {
        return studentList.get(index);
    }

//returns the Student object with a specified id
// searches through the array
//and stopping by returning when a match is found
    public Student findById(String id) {
        for (Student s : studentList) {
            if (s.getId().equals(id)) {
                return s;
            }
        }
        return null;
    }

    // counts the number of people in a specified year
    // demonstrates making a count with arraylists
    public int getCountOfPeopleAtYear(int year) {
        int count = 0;
        for (Student s : studentList) {
            if (s.getYear() == year) {
                count++;
            }
        }
        return count;
    }

//works out how many people in each year,
//then creates and returns a report
    //
    // demonstrates calculating a frequency report
    // i.e. how often each year occurs
    // it uses the value of the year as an index
    public String getYearsFrequencyReport() {
//work out max year
        int maxYear = getMaxYear();
//work out how many people at each year
        int[] count = new int[maxYear + 1];
        for (Student s : studentList) {
            int year = s.getYear();
            count[year]++;
        }
        String report = "";
        for (int year = 1; year <= maxYear; year++) {
            report += year + ": " + count[year] + "\n";
        }
        return report;
    }

//calculates the maximum year that anyone is in
//demonstrates finding a max with array lists
    public int getMaxYear() {
        int maxYear = 0;
        for (Student s : studentList) {
            int year = s.getYear();
            if (year > maxYear) {
                maxYear = year;
            }
        }
        return maxYear;
    }

    /**
     * writes supplied text to file
     *
     * @param filename the name of the file to be written to
     * @param report   the text to be written to the file
     */
    public void writeToFile(String filename, String report) {
        FileWriter fw;
        // catch the following exceptions FileNotFound, and IOException
        try {
            fw = new FileWriter(filename);
            fw.write(report);
            fw.close();
        } catch (FileNotFoundException e) {
            System.out.println("File not found: " + filename);
        } catch (IOException e) {
            System.out.println("IO Error: " + e.getMessage());
        }
    }

    /**
     * reads file with given name, extracting student data, creating student
```

```
objects
     * and adding them to the list of students Blank lines are skipped Validation
     * for integer year, missing items
     *
     * @param filename the name of the input file
     */
    public void readFile(String filename) {
        Scanner scan;
        try {
            scan = new Scanner(new File(filename));
            while (scan.hasNextLine()) {
                String line = scan.nextLine();
                if (!line.trim().equals("")) { // skip blank lines
                    processLine(line);
                }
            }
            scan.close();
        } catch (FileNotFoundException e) {
            System.out.println("File not found: " + filename);
        }
    }

    /**
     * Processes line, extracts data, creates Student object and adds to list
Checks
     * for non-numeric year and missing items Will still crash if name entered
     * without a space
     *
     * @param line - the line to be processed
     */
    private void processLine(String line) {
        try {
            String parts[] = line.split(",");
            Name name = new Name(parts[1]);
            String id = parts[0];
            String yearNum = parts[2];
            yearNum = yearNum.trim(); // remove any spaces
            int year = Integer.parseInt(yearNum);
//the qualifications are at the end of the line
            int qualLength = parts.length - 3;
            String quals[] = new String[qualLength];
            System.arraycopy(parts, 3, quals, 0, qualLength);
//create Student object and add to the list
            Student s = new Student(id, name, quals, year);
            this.add(s);
        }
//for these two formatting errors, ignore lines in error and try and carry on
//this catches trying to convert a String to an integer
        catch (NumberFormatException nfe) {
            String error = "Number conversion error in '" + line + "'-" +
nfe.getMessage();
            System.out.println(error);
        }
//this catches missing items if only one or two items
//other omissions will result in other errors
        catch (ArrayIndexOutOfBoundsException air) {
            String error = "Not enough items in  : '" + line + "' index position :
" + air.getMessage();
            System.out.println(error);
        }
```

```java
    }
}

```