StudentList.java

```
1package workshop5Src;
2
3
4import java.io.File;
11
12
13
14// using an ArrayList collect many students
15public class StudentList {
16
17                  private ArrayList<Student> studentList;
18
19              //create an empty arraylist
20              public StudentList() {
21              studentList = new ArrayList<Student>();
22
23          }
24          //add a student s to the collection
25          public void add(Student s) {
26
27
28                      studentList.add(s);
29
30
31          }
32
33                          //returns a report with one line per person
34                          // traverses the array list,
35                          //getting one element at a time
36                          public String getAllStudents()
37                          {
38                          String report = "ALL Students\n";
39                          for (Student  s : studentList){
40                          String fullName =  s.getName().getFullName();
41                          report += fullName + "\n";
42                          }
43                          report += "\n";
44
45              return report;
46              }
47
48              //returns the number of elements in the list
49              public int getSize() {
50              int TotalSize = studentList.size();
51
52                      return TotalSize;
53
54          }
55
56          //returns the Student object at specified index position
57          public Student getAtIndex(int index) {
58
59            return studentList.get(index);
60            }
61
62                              //returns the Student object with a specified id
```

StudentList.java

```
63                          // searches through the array
64                          //and stopping by returning when a match is found
65                          public Student findById(String id)
66                          {
67                          for (Student s : studentList) {
68                          if (s.getId().equals(id))
69                          return s;
70                          }
71
72          return null;
73          }
74
75
76
77          //counts the number of people in a specified year
78          //demonstrates making a count with arraylists 79    public int
            getCountOfPeopleAtYear(int year) {
80                          int count = 0 ;
81                          for (Student s: studentList) {
82                          if (s.getYear() == year)
83                          count+=1;
84                          }
85
86          return count;
87          }
88
89
90
91          //works out how many people in each year,
92          //then creates and returns a report
93          //
94          //demonstrates calculating a frequency report
95          //i.e. how often each year occurs
96          //it uses the value of the year as an index
97          public String getYearsFrequencyReport() {
98
99                  int count = 0;
100
101
102             int maxYear = getMaxYear();
103             //System.out.println(maxYear); 104    String report = "";
105                         for (Student s: studentList) {
106                         if (s.getYear() == maxYear) {
107                         count++;
108                         }
109                         report = "The frequency of "+ s.getYear()+" is " + count;
110                         }
111
112             //work out how many people at each year
113             //your code goes here
114             return report;
115             }
116
117                           //calculates the maximum year that anyone is in
118                           //demonstrates finding a max with array lists
119                           public int getMaxYear() {
```

StudentList.java

```
120                         int maxYear = studentList.get(0).getYear();
121                         for (int i = 1; i< studentList.size(); i++) {
122                         if (studentList.get(i).getYear() > maxYear) {
123                         maxYear = studentList.get(i).getYear();
124                         }
125
126             }
127             return maxYear;
128             }
129
130         /**
131        * writes supplied text to file 132      * @param filename the name of the file to be written to 133     * @param
report the text to be written to the file
134             */
135             public  void writeToFile(String filename, String report) {
136             try {
137             File file = new File(filename);
138             file.createNewFile();
139
140                     FileWriter fw = new FileWriter(file);
141                     fw.write(report);
142                     // fw.flush();
143                     fw.close();
144                     }catch(IOException e) {
145                     System.out.println("An error occured");
146                     e.printStackTrace();
147                     }
148
149
150// your code here
151//catch the following exceptions FileNotFound, and IOException 152
153         }
154
155       /** reads file with given name, extracting student data, creating student objects 156           * and adding them to the
list of students 157            * Blank lines are skipped 158          * Validation for integer year, missing items 159          *
@param filename the name of the input file
160             */
161             public void readFile(String filename) {
162             String data;
163             String id = null;       //unique id, e.g. 0099
164             String name = null;
165             int y = 0;       //year of study  e.g. 1,2,3 etc
166             String  [] pq = null;
167             int count = 0;
168
169
170                     //int count = 0;
171                     try {
172                     File file = new File(filename);
173                     Scanner myReader = new Scanner(file);
174                     while(myReader.hasNextLine()) { 175     data = myReader.nextLine();
176                                                     String[] parts = data.split(",");
177                                                     // System.out.println("The length is " + parts.length);
178                                                     for(int i = 0; i < parts.length; i++) {
179                                                     if (i == 0) {
180                                                     id = parts[i];
```

StudentList.java

```
181                                                 //System.out.println(id);
182                                                 }else if (i == 1) {
183                                                 name = parts[i];
184                                                 //System.out.println( name );
185                                                 }else if (i == 2) {
186                                                 count = 0;
187                                                 for (char c : parts[2].toCharArray()) {
188                                                 if(Character.isLetter(c))
189                                                 {
190                                                 count ++;
191                                                 }
192                                                 if (count == 0) {
193                                                 y =  Integer.parseInt(parts[2]);
194                                                 //System.out.println(y );
195                                                 }if (count > 0) {
196                                                 y = 0;
197
198                                         }
199
200
201                                 }
202
203                                     }else if(i == 3) {
204                                     pq = new String[]{parts[3]};
205                                     //System.out.println( pq);
206                                     }else if(i == 4) {
207                                     pq = new String[]{parts[4]};
208                                     //System.out.println(pq);
209                                     }
210
211
212
213
214                             }
215                             //System.out.println(id +' '+' '+ name +' '+ y + ' '+ pq);
216                             Name n = new Name(name);
217                             Student s = new Student(id, n, pq, y);
218                             studentList.add(s);
219                             }
220
221
222
223
224
225                         myReader.close();
226
227                     }catch(FileNotFoundException e) {
228                     System.out.println("An errorr occures");
229                     e.printStackTrace();
230
231             }
232
233             //your code here
234             }
235
236         /**
```

StudentList.java

```
237        * Processes line, extracts data, creates Student object 238        * and adds to list 239        * Checks for non-
numeric year and missing items 240             * Will still crash if name entered without a space 241 * @param line - the line
to be processed
242                     */
243                     private void processLine(String line) {
244                     try {
245                     String parts [] = line.split(",");
246                     Name name = new Name(parts[1]);
247                     String id = parts[0];
248                     String yearNum = parts[2];
249                     yearNum = yearNum.trim();  //remove any spaces
250                     int year = Integer.parseInt(yearNum);
251
252                     //the qualifications are at the end of the line
253                     int qualLength = parts.length - 3;
254                     String quals[] = new String[qualLength];
255                     System.arraycopy(parts, 3, quals, 0, qualLength); 256
257                     //create Student object and add to the list
258                     Student s = new Student(id,name, quals, year);
259                     this.add(s);
260                     }
261
262       //for these two formatting errors, ignore lines in error   and try and carry on 263
264                                                 //this catches trying to convert a String to an integer
265                                                 catch (NumberFormatException nfe) {
266                                                 String error = "Number conversion error in '" + line + "'                - "
267                                                 + nfe.getMessage();
268                                                 System.out.println(error);
269                                                 }
270                                                 //this catches missing items if only one or two items
271                                                 //other omissions will result in other errors
272                                                 catch (ArrayIndexOutOfBoundsException air) {
273                                                 String error = "Not enough items in  : '" + line
274                                                 + "' index position : " + air.getMessage();
275                                                 System.out.println(error);
276                                                 }
277
278         }
279}
280
```