

```
1  import java.io.Serializable;
2  import java.time.LocalDate;
3  import java.time.format.*;
4  import java.util.*;
5
6  public class Student implements Serializable
7  {
8      private String sNum;
9      private String name;
10     private String password;
11     private String phone;
12     private String address;
13     private LocalDate dateOfBirth;
14     private ArrayList<CourseEnrolment> currentEnrolments;
15     private ArrayList<Course> completedCourses;
16     private ArrayList<Course> exemptCourses;
17     static DateTimeFormatter dateFormat = DateTimeFormatter
18         .ofPattern("d/M/yyyy");
19     static DateTimeFormatter timeFormat = DateTimeFormatter.ofPattern("h:mm a");
20
21     public Student(String sNum, String name, String password, String phone,
22         String address, LocalDate dateOfBirth,
23         ArrayList<Course> exemptCourses)
24     {
25         this.sNum = sNum;
26         this.name = name;
27         this.password = password;
28         this.phone = phone;
29         this.address = address;
30         this.dateOfBirth = dateOfBirth;
31         this.currentEnrolments = new ArrayList<CourseEnrolment>();
32         this.completedCourses = new ArrayList<Course>();
33         this.exemptCourses = exemptCourses;
34     }
35
36     public CourseEnrolment enrol(CourseOffering offering)
37         throws OverEnrolmentException, AlreadyEnrolledException,
38         IncompletePrerequisitesException
39     {
40         // Check if the student is already enrolled in 4 courses for this period
41         if(currentEnrolments.size() >= 4)
42         {
43             throw new OverEnrolmentException("You are already enrolled in 4 "
44                 + "courses");
45         }
46
47         // Check if the student is already enrolled in this course
48         if(alreadyEnrolled(offering.getCourse()))
49         {
50             throw new AlreadyEnrolledException("You are already enrolled in "
51                 + "this course");
52         }
53
54         // Check if the student meets the prerequisites
55         if(!meetsPrerequisites(offering.getCourse()))
56         {
57             throw new IncompletePrerequisitesException("Prerequisites not met "
```

```
58         + "for this course");
59     }
60
61     CourseEnrolment newEnrolment = new CourseEnrolment(offering);
62     currentEnrolments.add(newEnrolment);
63
64     return newEnrolment;
65 }
66
67 public boolean alreadyEnrolled(Course course)
68 {
69     for(CourseEnrolment enrolment : currentEnrolments)
70     {
71         if(enrolment.getCourseOffering().getCourse().equals(course))
72         {
73             return true;
74         }
75     }
76     return false;
77 }
78
79 public boolean meetsPrerequisites(Course course)
80 {
81     for(Course prereq : course.getPrereqs())
82     {
83         if(!(exemptCourses.contains(prereq) || completedCourses
84             .contains(prereq)))
85         {
86             return false;
87         }
88     }
89     return true;
90 }
91
92 public void registerTutorial(CourseEnrolment courseEnrolment,
93     Tutorial tutorial) throws OverCapacityException,
94     TutorialAlreadyEnrolledException
95 {
96     if(tutorial.getNumberEnrolled() >= tutorial.getVenue().getCapacity())
97     {
98         throw new OverCapacityException("Venue at max capacity");
99     }
100
101     if(courseEnrolment.getTutorial() != null)
102     {
103         throw new TutorialAlreadyEnrolledException("You already have a "
104             + "tutorial registered for this course");
105     }
106
107     courseEnrolment.setTutorial(tutorial);
108
109     tutorial.setNumberEnrolled(tutorial.getNumberEnrolled() + 1);
110 }
111
112 public void deregisterTutorial(CourseEnrolment courseEnrolment)
113     throws TutorialNotEnrolledException
114 {
```

```
115         if(courseEnrolment.getTutorial() == null)
116         {
117             throw new TutorialNotEnrolledException("You are not enrolled in a "
118                 + "tutorial for this course");
119         }
120
121         // Reduce number enrolled in tutorial by 1
122         Tutorial tutorial = courseEnrolment.getTutorial();
123         tutorial.setNumberEnrolled(tutorial.getNumberEnrolled() - 1);
124
125         courseEnrolment.setTutorial(null);
126     }
127
128     public void printTimetable()
129     {
130         System.out.println("Name: " + name);
131
132         // Get maximum length for each field
133         int courseIDMaxLen = "Course".length();
134         int activityMaxLen = "Activity".length(); // "Tutorial" or "Lecture"
135         int locationMaxLen = "Location".length();
136         int dayMaxLen = "Day".length();
137         int startTimeMaxLen = "Start Time".length(); // Always longer/equal to
138         int endTimeMaxLen = "End Time".length(); // any time
139         int staffMaxLen = "Staff".length();
140         for(CourseEnrolment enrolment : currentEnrolments)
141         {
142             CourseOffering offering = enrolment.getCourseOffering();
143             Lecture lecture = offering.getLecture();
144             Tutorial tutorial = enrolment.getTutorial();
145
146             // Check if the length of this 'Course' field is largest so far
147             if(offering.getCourse().getID().length() > courseIDMaxLen)
148             {
149                 courseIDMaxLen = offering.getCourse().getID().length();
150             }
151
152             // Check if the length of this 'Location' field is largest so far
153             if(lecture.getVenue().getLocation().length() > locationMaxLen)
154             {
155                 locationMaxLen = lecture.getVenue().getLocation().length();
156             }
157             if(tutorial.getVenue().getLocation().length() > locationMaxLen)
158             {
159                 locationMaxLen = tutorial.getVenue().getLocation().length();
160             }
161
162             // Check if the length of this 'Day' field is largest so far
163             if(lecture.getDay().getDisplayName(TextStyle.FULL,
164                 Locale.getDefault()).length() > dayMaxLen)
165             {
166                 dayMaxLen = lecture.getDay().getDisplayName(TextStyle.FULL,
167                     Locale.getDefault()).length();
168             }
169             if(tutorial.getDay().getDisplayName(TextStyle.FULL,
170                 Locale.getDefault()).length() > dayMaxLen)
171             {
```

```

172         dayMaxLen = tutorial.getDay().getDisplay_name(TextStyle.FULL,
173             Locale.getDefault()).length();
174     }
175
176     // Check if the length of this 'Staff' field is largest so far
177     if(lecture.getStaff().getName().length() > staffMaxLen)
178     {
179         staffMaxLen = lecture.getStaff().getName().length();
180     }
181     if(tutorial.getStaff().getName().length() > staffMaxLen)
182     {
183         staffMaxLen = tutorial.getStaff().getName().length();
184     }
185 }
186 int totalLength = courseIDMaxLen + activityMaxLen + locationMaxLen
187     + dayMaxLen + startTimeMaxLen + endTimeMaxLen + staffMaxLen
188     + 20;
189
190 // Print top of table
191 System.out.println(String.format("%" + totalLength + "s", ""))
192     .replace(' ', '-'));
193 // Print table headers
194 System.out.println(String.format("| %-" + courseIDMaxLen + "s | %-"
195     + activityMaxLen + "s | %-" + locationMaxLen + "s | %-"
196     + dayMaxLen + "s | %-" + startTimeMaxLen + "s | %-"
197     + endTimeMaxLen + "s | %-" + staffMaxLen + "s |", "Course",
198     "Activity", "Location", "Day", "Start Time", "End Time",
199     "Staff"));
200 // Print header separator
201 System.out.println(String.format("| %-" + courseIDMaxLen + "s | %-"
202     + activityMaxLen + "s | %-" + locationMaxLen + "s | %-"
203     + dayMaxLen + "s | %-" + startTimeMaxLen + "s | %-"
204     + endTimeMaxLen + "s | %-" + staffMaxLen + "s |", "", "", "", "",
205     "", "", "", "").replace(' ', '-'));
206
207 // Print each table row
208 for(CourseEnrolment enrolment : currentEnrolments)
209 {
210     CourseOffering offering = enrolment.getCourseOffering();
211     Lecture lecture = offering.getLecture();
212     Tutorial tutorial = enrolment.getTutorial();
213
214     // Print lecture information
215     String s = "| ";
216     s += String.format("%-" + courseIDMaxLen + "s",
217         offering.getCourse().getID());
218     s += String.format(" | %-" + activityMaxLen + "s |", "Lecture");
219     s += String.format("%-" + locationMaxLen + "s",
220         lecture.getVenue().getLocation());
221     s += " | ";
222     s += String.format("%-" + dayMaxLen + "s", lecture.getDay()
223         .getDisplay_name(TextStyle.FULL, Locale.getDefault()));
224     s += " | ";
225     s += String.format("%-" + startTimeMaxLen + "s",
226         String.format("%8s",
227             timeFormat.format(lecture.getStart())));
228     s += " | ";

```

```
229         s += String.format("%-" + endTimeMaxLen + "s",
230             String.format("%8s", timeFormat.format(lecture.getEnd())));
231         s += " | ";
232         s += String.format("%-" + staffMaxLen + "s",
233             lecture.getStaff().getName());
234         s += " | ";
235         System.out.println(s);
236
237         // Print tutorial information
238         s = " | ";
239         s += String.format("%-" + courseIDMaxLen + "s",
240             offering.getCourse().getID());
241         s += String.format(" | %" + activityMaxLen + "s | ", "Tutorial");
242         s += String.format("%-" + locationMaxLen + "s",
243             tutorial.getVenue().getLocation());
244         s += " | ";
245         s += String.format("%-" + dayMaxLen + "s", tutorial.getDay()
246             .getDisplayName(TextStyle.FULL, Locale.getDefault()));
247         s += " | ";
248         s += String.format("%-" + startTimeMaxLen + "s",
249             String.format("%8s",
250                 timeFormat.format(tutorial.getStart())));
251         s += " | ";
252         s += String.format("%-" + endTimeMaxLen + "s",
253             String.format("%8s", timeFormat.format(tutorial.getEnd())));
254         s += " | ";
255         s += String.format("%-" + staffMaxLen + "s",
256             tutorial.getStaff().getName());
257         s += " | ";
258         System.out.println(s);
259     }
260
261     // Print bottom of table
262     System.out.println(String.format("+%" + totalLength + "s+", "")
263         .replace(' ', '-'));
264 }
265
266 public String getPassword()
267 {
268     return password;
269 }
270
271 public void setPassword(String password)
272 {
273     this.password = password;
274 }
275
276 public String getSNum()
277 {
278     return sNum;
279 }
280
281 public String getName()
282 {
283     return name;
284 }
285
```

```
286     public String getPhone()
287     {
288         return phone;
289     }
290
291     public ArrayList<CourseEnrolment> getCurrentEnrolments()
292     {
293         return currentEnrolments;
294     }
295
296     public String[] getCurrentEnrolmentNames()
297     {
298         ArrayList<String> enrolmentNames = new ArrayList<String>();
299         for(CourseEnrolment enrolment : currentEnrolments)
300         {
301             enrolmentNames.add("[ " + enrolment.getCourseOffering().getCourse()
302                                 .getID() + " ] " + enrolment.getCourseOffering().getCourse()
303                                 .getName());
304         }
305         return enrolmentNames.toArray(new String[currentEnrolments.size()]);
306     }
307
308     public ArrayList<Course> getCompletedCourses()
309     {
310         return completedCourses;
311     }
312
313     public ArrayList<Course> getExemptCourses()
314     {
315         return exemptCourses;
316     }
317
318     public String toString()
319     {
320         String s = "Student ID: " + sNum;
321         s += "\nName" + name;
322         s += "\nPhone Number: " + phone;
323         s += "\nAddress: " + address;
324         s += "\nDOB: " + dateFormat.format(dateOfBirth);
325         s += "\nCurrent enrolments: ";
326
327         String delim = "";
328         for(CourseEnrolment offering : currentEnrolments)
329         {
330             s += delim + offering.getCourseOffering().getCourse().getID();
331             s += " - " + offering.getCourseOffering().getCourse().getName();
332             delim = ", ";
333         }
334
335         return s;
336     }
337 }
338
```