```java
 1 package entity;
 2
 3 import java.io.Serializable;
 4 import java.time.LocalDate;
 5 import java.util.Objects;
 6
 7 /**
 8  *
 9  * @author Chew Lip Sin
10  */
11 public class Course implements Serializable,Comparable<Course> {
12
13     private String courseCode;
14     private String title;
15     private int creditHours;
16     private Sem semester;
17     private LocalDate createdAt;
18     private LocalDate updatedAt;
19
20     @Override
21     public int compareTo(Course o) {
22         return courseCode.compareTo(o.courseCode);
23     }
24
25
26     /**
27      * Enumeration for different intake semesters.
28      */
29     public enum Sem {
30         /**
31          * Represents all semesters.
32          */
33         ALL,
34         /**
35          * Represents the January semester.
36          */
37         JAN,
38         /**
39          * Represents the July semester.
40          */
41         JUL;
42
43         /**
44          * Returns the string representation of a semester.
45          *
46          * @param sem The semester to convert to a string.
```

```java
47              * @return The string representation of the semester.
48             */
49          public String getString(Sem sem) {
50              switch (sem) {
51                  case ALL:
52                      return "ALL";
53                  case JAN:
54                      return "JAN";
55                  case JUL:
56                      return "JUL";
57              }
58              return "ALL";
59          }
60
61      };
62
63      /**
64       * Constructs a new Course instance with specified details.
65       *
66       * @param courseCode The code of the course.
67       * @param title The title of the course.
68       * @param creditHours The credit hours of the course.
69       * @param semester The intake semester of the course.
70       */
71      public Course(String courseCode, String title, int creditHours,
72              Sem semester) {
73          this.courseCode = courseCode;
74          this.title = title;
75          this.creditHours = creditHours;
76          this.semester = semester;
77          this.createdAt = LocalDate.now();
78          this.updatedAt = LocalDate.now();
79      }
80
81      /**
82       * Default constructor for the Course class.
83       */
84      public Course() {
85      }
86
87      /**
88       * Returns the course code.
89       *
90       * @return The course code.
91       */
92      public String getCourseCode() {
```

```java
 93            return courseCode;
 94        }
 95
 96        /**
 97         * Returns the title of the course.
 98         *
 99         * @return The title of the course.
100         */
101        public String getTitle() {
102            return title;
103        }
104
105        /**
106         * Returns the credit hours of the course.
107         *
108         * @return The credit hours of the course.
109         */
110        public int getCreditHours() {
111            return creditHours;
112        }
113
114        /**
115         * Updates the updatedAt date of the course to the current date.
116         */
117        public void update() {
118            this.updatedAt = LocalDate.now();
119        }
120
121        /**
122         * Sets the course code of the course.
123         *
124         * @param courseCode The new course code to set.
125         */
126        public void setCourseCode(String courseCode) {
127            this.courseCode = courseCode;
128        }
129
130        /**
131         * Gets the date when the course was created.
132         *
133         * @return The creation date of the course.
134         */
135        public LocalDate getCreatedAt() {
136            return createdAt;
137        }
138
```

2023.09.02 02:08:52

```
139        /**
140         * Gets the date when the course was last updated.
141         *
142         * @return The last updated date of the course.
143         */
144        public LocalDate getUpdatedAt() {
145            return updatedAt;
146        }
147
148        /**
149         * Sets the date when the course was last updated.
150         *
151         * @param updatedAt The updated date to set.
152         */
153        public void setUpdatedAt(LocalDate updatedAt) {
154            this.updatedAt = updatedAt;
155        }
156
157        /**
158         * Sets the date when the course was created.
159         *
160         * @param createdAt The creation date to set.
161         */
162        public void setCreatedAt(LocalDate createdAt) {
163            this.createdAt = createdAt;
164        }
165
166        /**
167         * Sets the title of the course.
168         *
169         * @param title The new title to set.
170         */
171        public void setTitle(String title) {
172            this.title = title;
173        }
174
175        /**
176         * Sets the credit hours of the course.
177         *
178         * @param creditHours The new credit hours to set.
179         */
180        public void setCreditHours(int creditHours) {
181            this.creditHours = creditHours;
182        }
183
184        /**
```

```java
185          * Computes and returns the hash code value for this course.
186          *
187          * @return The hash code value for this course.
188          */
189         @Override
190         public int hashCode() {
191             int hash = 3;
192             hash = 59 * hash + Objects.hashCode(this.courseCode);
193             hash = 59 * hash + Objects.hashCode(this.title);
194             hash = 59 * hash + this.creditHours;
195             return hash;
196         }
197
198         /**
199          * Indicates whether some other object is "equal to" this course.
200          *
201          * @param obj The reference object with which to compare.
202          * @return {@code true} if this course is the same as the obj
203          * argument;{@code false} otherwise.
204          */
205         @Override
206         public boolean equals(Object obj) {
207             if (this == obj) {
208                 return true;
209             }
210             if (obj == null) {
211                 return false;
212             }
213             if (getClass() != obj.getClass()) {
214                 return false;
215             }
216             final Course other = (Course) obj;
217             if (this.creditHours != other.creditHours) {
218                 return false;
219             }
220             if (!Objects.equals(this.courseCode, other.courseCode)) {
221                 return false;
222             }
223             return Objects.equals(this.title, other.title);
224         }
225
226         /**
227          * Returns the semester of the course.
228          *
229          * @return The semester of the course.
230          */
```

```java
231        public Sem getSemester() {
232            return semester;
233        }
234
235        /**
236         * Sets the semester of the course.
237         *
238         * @param semester The new semester to set.
239         */
240        public void setSemester(Sem semester) {
241            this.semester = semester;
242        }
243
244        /**
245         * Compares this course's semester with another semester.
246         *
247         * @param sem The semester to compare against.
248         * @return A negative integer, zero, or a positive integer as this semester
249         * is less than, equal to, or greater than the specified semester.
250         */
251        public int compareSem(Sem sem) {
252            if (this.semester.compareTo(sem) < 0) {
253                return -1;
254            } else if (this.semester.compareTo(sem) == 0) {
255                return 0;
256            } else {
257                return 1;
258            }
259        }
260
261        /**
262         * Returns a string representation of the course's semester.
263         *
264         * @param semester The semester to convert to a string.
265         * @return The string representation of the semester.
266         */
267        public String semToString(Sem semester) {
268            if (semester == Sem.JAN) {
269                return "JAN";
270            } else if (semester == Sem.JUL) {
271                return "JUL";
272            } else if (semester == Sem.ALL) {
273                return "ALL";
274            } else {
275                return null;
276
```

```java
277                 }
278         }
279
280         /**
281          * Returns a string representation of the Course object.
282          *
283          * @return A formatted string representation of the Course object.
284          */
285         @Override
286         public String toString() {
287             String sems;
288             if (semester == Sem.JAN) {
289                 sems = "JAN";
290             } else if (semester == Sem.JUL) {
291                 sems = "JUL";
292             } else {
293                 sems = "ALL";
294             }
295             return String.format("|%-12s|%-52s|  %-2d            |  %-8s|%-12s|"
296                     + "%-12s", courseCode, title,
297                     creditHours, sems, createdAt, updatedAt);
298         }
299 }
```