

```
1  /*****
2  * Dalton Nofs
3  * Login ID: nofs5491
4  * CS-102, Summer 2017
5  * Programming Assignment 4
6  * CourseSearch class: used to search a database for a course *
7  *****/
8  public class CourseSearch
9  {
10     /*****
11     * Method: findByTitle()
12     * Purpose: Searches database for matching course number
13     *
14     * Parameters:
15     *     String: courseTitle: Course title to search for
16     *     Database: targetDatabase: database to be searched
17     * Returns:
18     *     Void: nothing to be returned
19     *****/
20     public void findByTitle(String courseTitle, Database targetDatabase) throws NoSuchFieldException
21     {
22         String buffer = ""; // Out buffer
23
24         // Search given database for a matching string
25         for(int index=0; index<targetDatabase.getDatabaseSize(); index++)
26         {
27             buffer += gather(courseTitle, targetDatabase.get(index).getRoot());
28         }
29         if(!buffer.equals(""))
30         {
31             System.out.println("Results:");
32             System.out.println(buffer);
33             buffer = ""; // clear buffer
34         }
35         // Throw exception as no match was found
36         else{throw new NoSuchFieldException();}
37     }
38
39     /*****
40     * Method: findByNumber()
41     * Purpose: Searches database for matching course number
42     *
43     * Parameters:
44     *     String: courseNumber: Course number to search for
45     *     Database: targetDatabase: database to be searched
46     * Returns:
47     *     LinkedList<Course> search results
48     *****/
49     public LinkedList<Course> findByNumber(String courseNumber, Database targetDatabase) throws NoSuchFieldException
50     {
51         String buffer = ""; // Out buffer
52         Course tempCourse = new Course(); // Temp course for searching tree
53         LinkedList<Course> returnBuff = new LinkedList<Course>(); // search results
54         tempCourse.setCourseNumber(courseNumber);
55
56         // Search given database for a matching string
57         for(int index=0; index<targetDatabase.getDatabaseSize(); index++)
58         {
59             if(targetDatabase.get(index).search(tempCourse))
60             {
61                 // Add the class attributes to the buffer
62                 buffer += targetDatabase.get(index).getSearched().getCourseNumber()
63                     + ": " +
64                     targetDatabase.get(index).getSearched().getCourseTitle()
65                     + " (" +
66                     targetDatabase.get(index).getSearched().getCreditCount()
67                     + ") . " +
68                     targetDatabase.get(index).getSearched().getTermTaken()
69                     + " " +
70                     targetDatabase.get(index).getSearched().getYearTaken()
71                     + " " +
72                     targetDatabase.get(index).getSearched().getCourseGrade()
73                     + "\n" ;
74                 // add last searched to LinkedList of found items
75                 returnBuff.addLast(targetDatabase.get(index).getSearched());
76             }
77         }
78         if(!buffer.equals(""))
79         {
80             System.out.println("Results:");
```

```
81         System.out.println(buffer);
82         buffer = ""; // clear buffer
83         return(returnBuff);
84     }
85     // Throw exception as no match was found
86     else{throw new NoSuchFieldException();}
87 }
88
89 /*****
90  * Method: gather() *private*
91  * Purpose: fills buffer with tree
92  *
93  * Parameters: String: TreeNode: target, current node
94  * Returns: String: compiled buffer from tree
95  *****/
96 private String gather(String target, TreeNode<Course> current)
97 {
98     String buffer = "";
99     if(current == null) {return buffer;} // if fallen off list
100
101     if(current.getDatum().getCourseTitle().
102         toLowerCase().contains(target.toLowerCase()))
103     {
104         // Add the class attributes to the buffer
105         buffer += current.getDatum().getCourseNumber() + ": " +
106             current.getDatum().getCourseTitle() + " (" +
107             current.getDatum().getCreditCount() + "). " +
108             current.getDatum().getTermTaken() + " " +
109             current.getDatum().getYearTaken() + " " +
110             current.getDatum().getCourseGrade();
111         // Check to see if excluded from GPA calc
112         if(current.getDatum().getExcludeFlag().equals("Y") ||
113             current.getDatum().getExcludeFlag().equals("y"))
114         {
115             buffer += " (excluded).\n";
116         }
117         else{buffer += ".\n";}
118     }
119     // gather the rest of the left till null
120     buffer += gather(target,current.getRight());
121     // gather the rest of the right till null
122     buffer += gather(target,current.getLeft());
123     return buffer;
124 }
125
126 }
```