```java
/************************************************************
* Dalton Nofs                                               *
* Login ID: nofs5491                                        *
* CS-102, Summer 2017                                       *
* Programming Assignment 5                                  *
* CourseSearch class:  used to search a database for a course *
************************************************************/
public class CourseSearch
{
    /************************************************************
    * Method: findByTitle()                                      *
    * Purpose: Searches database for matching course number      *
    *                                                             *
    * Parameters:                                                *
    *         String: courseTitle: Course title to search for     *
    *         Database: targetDatabase: database to be searched   *
    * Returns:                                                    *
    *         Void: nothing to be returned                       *
    ************************************************************/
    public void findByTitle(String courseTitle, Database targetDatabase) throws NoSuchFieldException
    {
        String buffer = ""; // Out buffer

        // Search given database for a matching string
        for(int index=0; index<targetDatabase.getDatabaseSize(); index++)
        {
            buffer += gather(courseTitle, targetDatabase.get(index).getRoot());
        }
        if(!buffer.equals(""))
        {
            UserInterface.sendMessage(buffer, "Results");
            buffer = ""; // clear buffer
        }
        // Throw exception as no match was found
        else{throw new NoSuchFieldException();}
    }

    /************************************************************
    * Method: findByNumber()                                    *
    * Purpose: Searches database for matching course number      *
    *                                                             *
    * Parameters:                                                *
    *         String: courseNumber: Course number to search for   *
    *         Database: targetDatabase: database to be searched   *
    * Returns:                                                    *
    *         LinkedList<Course> search results                   *
    ************************************************************/
    public LinkedList<Course> findByNumber(String courseNumber, Database targetDatabase) throws NoSuchFieldException
    {
        String buffer = ""; // Out buffer
        Course tempCourse = new Course(); // Temp course for searching tree
        LinkedList<Course> returnBuff = new LinkedList<Course>(); // search results
        tempCourse.setCourseNumber(courseNumber);

        // Search given database for a matching string
        for(int index=0; index<targetDatabase.getDatabaseSize(); index++)
        {
            if(targetDatabase.get(index).search(tempCourse))
            {
                // Add the class attributes to the buffer
                buffer += targetDatabase.get(index).getSearched().getCourseNumber()
                          + ": " +
                          targetDatabase.get(index).getSearched().getCourseTitle()
                              + " (" +
                          targetDatabase.get(index).getSearched().getCreditCount()
                              + "). "+
                          targetDatabase.get(index).getSearched().getTermTaken()
                              + " "  +
                          targetDatabase.get(index).getSearched().getYearTaken()
                              + " "  +
                          targetDatabase.get(index).getSearched().getCourseGrade()
                              + "\n"    ;
                // add last searched to LinkedList of found items
                returnBuff.addLast(targetDatabase.get(index).getSearched());
            }
        }
        if(!buffer.equals(""))
        {
            UserInterface.sendMessage(buffer, "Results");
            buffer = ""; // clear buffer
```

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