```java
 1  /***************************************************************
 2   * Dalton Nofs                                                 *
 3   * Login ID: nofs5491                                          *
 4   * CS-102, Summer 2017                                         *
 5   * Programming Assignment 4                                    *
 6   * ConsolePrint class:  general use program printer            *
 7   ***************************************************************/
 8
 9  public class ConsolePrint
10  {
11      /***************************************************************
12       * Method: printDatabase()                                     *
13       * Purpose: writes out the entire database                     *
14       *                                                             *
15       * Parameters:   Database: printBase: database to be printed   *
16       * Returns: Void:           nothing will be returned           *
17       ***************************************************************/
18      public void printDatabase(Database printBase) throws IllegalArgumentException
19      {
20          String buffer = ""; // Buffer string for data to be sent to console
21          int arrayCount = 0; // local arraycount for cycling through the database
22          arrayCount = printBase.getDatabaseSize();
23
24          // Check the status of the database
25          if(arrayCount <= 0)
26          {
27              throw new IllegalArgumentException("Database is empty!\n");
28          }
29
30          // Loop semesters
31          for(int index=0; index<arrayCount; index++)
32          {
33              buffer += gather(printBase.get(index).getRoot());
34          }
35          System.out.println(buffer); // Print the entire database
36          buffer = ""; // clear the buffer
37      }
38
39      /***************************************************************
40       * Method: gather()     *private*                              *
41       * Purpose: fills buffer with tree                             *
42       *                                                             *
43       * Parameters: TreeNode:        current node                   *
44       * Returns: String:             compiled buffer from tree      *
45       ***************************************************************/
46      private String gather(TreeNode<Course> current)
47      {
48          String buffer = "";
49          if(current == null) {return buffer;} // if fallen off list
50          // gather the rest of the right till null
51          buffer += gather(current.getRight());
52
53          buffer += current.getDatum().getCourseNumber() + ": " +
54                    current.getDatum().getCourseTitle()  + " (" +
55                    current.getDatum().getCreditCount()  + "). "+
56                    current.getDatum().getTermTaken()       + " "  +
57                    current.getDatum().getYearTaken()       + " "  +
58                    current.getDatum().getCourseGrade()  + ""    ;
59          // Check to see if excluded from GPA calc
60          if(current.getDatum().getExcludeFlag().equals("Y") ||
61             current.getDatum().getExcludeFlag().equals("y"))
62          {
63              buffer += " (excluded).\n";
64          }
65          else{buffer += ".\n";}
66          // gather the rest of the left till null
67          buffer += gather(current.getLeft());
```

```
68            return buffer;
69        }
70
71 }
```