```java
import java.text.ParseException;
import java.util.InputMismatchException;
import java.util.LinkedList;
import java.util.Scanner;

/*************************************************************
* Dalton Nofs                                                *
* Login ID: nofs5491                                         *
* CS-102, Summer 2017                                        *
* Programming Assignment 3                                   *
* Assignment class:  main entry point for assignment 3       *
*************************************************************/
public class Prog3
{
    public static void main (String[] args)
    {
        ConsolePrint localPrinter = new ConsolePrint();     // local printer for general printing
        CourseSearch courseSearcher = new CourseSearch();     // Database searcher for course's
        GpaCalc gpaCalculator = new GpaCalc();                // Calcualtor for gpa via database
        Scanner console = new Scanner(System.in); // Scanner for parsing the console
        int userInput = 0;                        // User input for selecting option
        Database courseData = new Database();     // Create a database to import to
        // String for printing the greeting message
        String greetingString = "Welcome to the CS-102 Transcript Program" +
                                "\nCurrent available commands:"            +
                                "\n    1 --> Search for a course number"   +
                                "\n    2 --> Search course titles"         +
                                "\n    3 --> Print all records"            +
                                "\n    4 --> Computer GPAs"                +
                                "\n    5 --> Add Course"                   +
                                "\n    6 --> Remove Course"                   +
                                "\n    7 --> Edit Course"                  +
                                "\n    9 --> Exit"                           ;

        final int choice1 = 1; // Course by number
        final int choice2 = 2; // Course by title
        final int choice3 = 3; // Print all records
        final int choice4 = 4; // Computer GPAs
        final int choice5 = 5; // Add a course
        final int choice6 = 6; // Remove a course
        final int choice7 = 7; // Edit a course
        final int choice9 = 9; // Exit
        try
        {
            // attempt to load the database with file location
            //   given at runtime
            courseData.loadDatabase(args);
        }
        catch(ArrayIndexOutOfBoundsException exc)
        {
            System.out.println("The database is full, " + exc.getMessage() + " course(s) were loaded!");
        }
        catch(IllegalArgumentException exc)
        {
            System.out.println(exc.getMessage());
        }

        System.out.println("Database has been loaded!\n");

        while(true)
        {
            System.out.println(greetingString); // Print welcome message
            try
            {
                userInput = console.nextInt();
                console.nextLine(); // Clear the scanner's buffer by going to the return char
                                    //    still viable if a newline char is pasted into terminal
            }
            catch(InputMismatchException exc)
            {
                // Just change the user input to use the default catch
                userInput = 0;
            }

            switch (userInput)
            {
                case choice1: System.out.println("What is the Number of the course?: ");
```

```
78                      try
79                      {
80                          courseSearcher.findByNumber(console.next(), courseData);
81                      }
82                      catch(NoSuchFieldException exc)
83                      {
84                          System.out.println("Course was not found!\n");
85                      }
86                      break;
87
88              case choice2: System.out.println("What is the title of the course?: ");
89                      try
90                      {
91                          courseSearcher.findByTitle(console.next(), courseData);
92                      }
93                      catch(NoSuchFieldException exc)
94                      {
95                          System.out.println("Course was not found!\n");
96                      }
97                      break;
98
99              case choice3: System.out.println("Printing all records\n");
100                     try
101                     {
102                         localPrinter.printDatabase(courseData);
103                     }
104                     catch(IllegalArgumentException exc)
105                     {
106                         System.out.println(exc.getMessage());
107                     }
108                     break;
109
110             case choice4: System.out.print("Students GPA: ");
111                     try
112                     {
113                         System.out.format("%.2f\n\n",gpaCalculator.calcGpa(courseData));
114                     }
115                     catch(IllegalArgumentException exc)
116                     {
117                         System.out.println(exc.getMessage());
118                     }
119                     break;
120
121             case choice5: userAddCourse(courseData, console);
122                           break;
123
124             case choice6: userRemoveCourse(courseData, console, courseSearcher);
125                           break;
126
127             case choice7: userEditCourse(courseData, console, courseSearcher);
128                           break;
129
130             case choice9: System.out.println("Exiting");
131                           System.exit(0); // exit successfully
132
133             // Catch all (commands that are not 1-4,9)
134             default: System.out.println("The command you entered is not recognized.\n");
135                      break;
136         }
137     }
138 }
139
140 /************************************************************
141 * Method: userAddClass()                                    *
142 * Purpose: add a user class to database                     *
143 *                                                              *
144 * Parameters: Database: Scanner: targetbase, console scanner *
145 * Returns: void:                  N/A                       *
146 ************************************************************/
147 private static void userAddCourse(Database targetDatabase, Scanner console)
148 {
149     System.out.println("Enter your class in the following format:\n"  +
150                        "201003/CE-320/4/Microcomputers I/B+/N\n"      +
151                        "yyyytt/corsnum/credit/title/grade/excluded\n" +
152                        "yyyy is year tt is term (01,02,03,04)     "  );
153     String userInput = ""; // User input string to be feed to addCourse
154     Course tempCourse = new Course(); // course to be added
155     String dateString = ""; // Temp string for separating the year and semester
```

```
156         userInput = console.nextLine(); // get users input
157         Scanner pieces = new Scanner(userInput); // Scanner for spliting string
158 
159         // setTermTaken, and setExcludeFlag will throw a parse error
160         //     if data sent is not in the correct format
161         try
162         {
163             pieces.useDelimiter("/");
164             dateString = pieces.next();
165             if(dateString.length() != 6)
166             {
167                 throw new ParseException("Year/Term is wrong length",0);
168             }
169             tempCourse.setYearTaken(dateString.substring(0, 4)); // Set year to string char's 0-4 (year)
170             tempCourse.setTermTaken(dateString.substring(4, 6)); // Set term taken to the 2 digit semester code
171             tempCourse.setCourseNumber(pieces.next());             // Set the course number
172             tempCourse.setCreditCount(pieces.nextInt());         // Set the number of credits the class is worth
173             tempCourse.setCourseTitle(pieces.next());             // Set the course title
174             tempCourse.setCourseGrade(pieces.next().toUpperCase()); // Set the course grade
175             tempCourse.setExcludeFlag(pieces.next());             // Set the exclude flag
176         }
177         catch(ParseException exc)
178         {
179             System.out.println(exc.getMessage() + " Your input is ignored!\n");
180             return;
181         }
182         catch(InputMismatchException exc)
183         {
184             System.out.println(exc.getMessage()+"your input is ignored\n");
185             return;
186         }
187         targetDatabase.addCourse(tempCourse);
188         System.out.println("\n"); // extra space for prettyness
189     }
190 
191     /*************************************************************
192     * Method: userRemoveCourse()                                 *
193     * Purpose: remove a user class to database                      *
194     *                                                               *
195     * Parameters: Database: Scanner: targetbase, console scanner *
196     * Returns: void:                    N/A                      *
197     *************************************************************/
198     private static void userRemoveCourse(Database targetDatabase, Scanner console,
199             CourseSearch courseSearcher)
200     {
201         int promptCtr = 0; // Counter for seeing how many time the user was prompted
202         System.out.println("Please enter the course number you would like to remove:");
203         String userInput = console.next(); // users course input
204         console.nextLine(); // Clear the scanner's buffer by going to the return char
205                             //    still viable if a newline char is pasted into terminal
206         LinkedList<Integer> returnResults; // Results from the course search
207         int deletedCounter = 0; // counter for number of courses deleted
208         try
209         {
210             System.out.print("Course search, ");
211             returnResults = courseSearcher.findByNumber(userInput, targetDatabase);
212         }
213         catch (NoSuchFieldException exc)
214         {
215             System.out.println("No results were found!\n");
216             return;
217         }
218         System.out.println("Please enter the term you would like to remove it from " +
219                            "(yyyytt): ");
220         String termInput = console.next(); // capture the term to delete from
221         console.nextLine(); // Clear the scanner's buffer by going to the return char
222                             //     still viable if a newline char is pasted into terminal
223         // return results are stored resut[1] = index 1, reult[2] index2
224         for(int index=0;index<returnResults.size();)
225         {
226             // Print the course as long as it matches year and term
227             if(termInput.equalsIgnoreCase(targetDatabase.getArrayPosition(
228                     ((int) returnResults.get(index)),
229                     ((int)returnResults.get(index+1))).getYearTaken()
230                     +
231                     targetDatabase.getArrayPosition(
232                     ((int) returnResults.get(index)),
233                     ((int)returnResults.get(index+1))).getTermTakenRaw())
```

```
234                 )
235             {
236                 printCourse(targetDatabase, returnResults, index);
237                 System.out.println("Would you like to delete(y/n):");
238                 userInput = console.next();
239                 if(userInput.equalsIgnoreCase("y"))
240                 {
241                     // remove course from lower list
242                     targetDatabase.remove(((int) returnResults.get(index)),
243                             ((int)returnResults.get(index+1)));
244                     deletedCounter++;
245                 }
246                 else{/* do nothing */}
247                 promptCtr++;
248             }
249             index += 2; // because of storage in results add 2 instead of 1
250         }
251         if(promptCtr>0)
252             System.out.println("You deleted " + deletedCounter + " course(s)!\n");
253         else
254             System.out.println("There were no courses with the specified term!\n");
255     }
256
257     /************************************************************
258     * Method: userRemoveCourse()                                *
259     * Purpose: remove a user class to database                     *
260     *                                                              *
261     * Parameters: Database: Scanner: targetbase, console,          *
262     * Returns: void:                    N/A                        *
263     ************************************************************/
264     private static void userEditCourse(Database targetDatabase, Scanner console, CourseSearch courseSearcher)
265     {
266         int promptCtr = 0; // Counter for seeing how many time the user was prompted
267         System.out.println("Please enter the course number you would like to edit:");
268         String userInput = console.next(); // users course input
269         console.nextLine(); // Clear the scanner's buffer by going to the return char
270                             //    still viable if a newline char is pasted into terminal
271         LinkedList<Integer> returnResults; // Results from the course search
272         int editedCounter = 0; // For the number of edited courses
273         try
274         {
275             System.out.print("Course search, ");
276             returnResults = courseSearcher.findByNumber(userInput, targetDatabase);
277         }
278         catch (NoSuchFieldException e)
279         {
280             System.out.println("No results were found!\n");
281             return;
282         }
283         System.out.println("Please enter the term you would like to edit it in " +
284                            "(yyyytt): ");
285         String termInput = console.next(); // capture the term to delete from
286         console.nextLine(); // Clear the scanner's buffer by going to the return char
287                             //     still viable if a newline char is pasted into terminal
288         // return results are stored resut[1] = index 1, reult[2] index2
289         for(int index=0;index<returnResults.size();)
290         {
291             // Print the course as long as it matches year and term
292             if(termInput.equalsIgnoreCase(targetDatabase.getArrayPosition(
293                     ((int) returnResults.get(index)),
294                     ((int)returnResults.get(index+1))).getYearTaken()
295                     +
296                     targetDatabase.getArrayPosition(
297                     ((int) returnResults.get(index)),
298                     ((int)returnResults.get(index+1))).getTermTakenRaw())
299                 )
300             {
301                 printCourse(targetDatabase, returnResults, index);
302                 System.out.println("Would you like to Edit?(y/n):");
303                 userInput = console.next();
304                 if(userInput.equalsIgnoreCase("y"))
305                 {
306                     // remove course from lower list then re add the modified
307                     targetDatabase.remove(((int) returnResults.get(index)),
308                             ((int)returnResults.get(index+1)));
309                     // create a new scanner as the old one causes problems
310                     userAddCourse(targetDatabase, new Scanner(System.in));
311                     editedCounter++;
```

```
312                 }
313                 else{/* do nothing */}
314                 promptCtr++;
315             }
316             index += 2; // because of storage in results add 2 instead of 1
317         }
318         if(promptCtr>0)
319             System.out.println("You edited " + editedCounter + " course(s)!\n");
320         else
321             System.out.println("There were no courses with the specified term!\n");
322     }
323 
324     /************************************************************
325     * Method: printCourse()                                      *
326     * Purpose: print a singular course                          *
327     *                                                          *
328     * NOTE: this was making all my methods too long so...      *
329     *                                                          *
330     * Parameters: Database: LinkedList: Index:                  *
331     *                 targetDatabase, returnResults, index      *
332     * Returns: void:                  N/A                       *
333     ************************************************************/
334     private static void printCourse(Database targetDatabase, LinkedList<Integer> returnResults, int index)
335     {
336         System.out.print(
337             targetDatabase.getArrayPosition(((int) returnResults.get(index)),
338                     ((int)returnResults.get(index+1))).getCourseNumber() + ": " +
339             targetDatabase.getArrayPosition(((int) returnResults.get(index)),
340                     ((int)returnResults.get(index+1))).getCourseTitle()  + " (" +
341             targetDatabase.getArrayPosition(((int) returnResults.get(index)),
342                     ((int)returnResults.get(index+1))).getCreditCount()  + "). "+
343               targetDatabase.getArrayPosition(((int) returnResults.get(index)),
344                       ((int)returnResults.get(index+1))).getTermTaken()     + " "  +
345               targetDatabase.getArrayPosition(((int) returnResults.get(index)),
346                       ((int)returnResults.get(index+1))).getYearTaken()     + " "  +
347               targetDatabase.getArrayPosition(((int) returnResults.get(index)),
348                       ((int)returnResults.get(index+1))).getCourseGrade()  + "\n"    );
349     }
350 }
```