```java
import java.io.BufferedWriter;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.OutputStreamWriter;
import java.io.Writer;
import java.text.ParseException;
import java.util.InputMismatchException;
import java.util.Scanner;

/************************************************************
* Dalton Nofs                                               *
* Login ID: nofs5491                                        *
* CS-102, Summer 2017                                       *
* Programming Assignment 4                                  *
* Assignment class:  main entry point for assignment 3      *
*************************************************************/
public class Prog4
{
    /************************************************************
    * Method: main()                                            *
    * Purpose: main entry for the program                         *
    *                                                               *
    * Parameters: String[]:             string array of prog args     *
    * Returns: void:                    N/A                         *
    *************************************************************/
    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 --> Compute GPA"                    +
                                "\n    5 --> Add Course"                   +
                                "\n    6 --> Remove Course"                    +
                                "\n    7 --> Edit Course"                  +
                                "\n    8 --> Store Database"               +
                                "\n    9 --> Reload Database"              +
                                "\n    0 --> Exit"                           ;

        final int srchCrsNum = 1; // Course by number
        final int srchTitle  = 2; // Course by title
        final int prntData   = 3; // Print all records
        final int cmptGpa    = 4; // Computer GPAs
        final int addCrs     = 5; // Add a course
        final int rmCrs      = 6; // Remove a course
        final int editCrs    = 7; // Edit a course
        final int strData    = 8; // Store Database
        final int reLdData   = 9; // Reload Database
        final int exit       = 0; // 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 srchCrsNum: System.out.println("What is the Number of the course?: ");
                        try
                        {
                            courseSearcher.findByNumber(console.next(), courseData);
                        }
                        catch(NoSuchFieldException exc)
                        {
                            System.out.println("Course was not found!\n");
                        }
                        break;

                case srchTitle: System.out.println("What is the title of the course?: ");
                        try
                        {
                            courseSearcher.findByTitle(console.next(), courseData);
                        }
                        catch(NoSuchFieldException exc)
                        {
                            System.out.println("Course was not found!\n");
                        }
                        break;

                case prntData: System.out.println("Printing all records\n");
                        try
                        {
                            localPrinter.printDatabase(courseData);
                        }
                        catch(IllegalArgumentException exc)
                        {
                            System.out.println(exc.getMessage());
                        }
                        break;

                case cmptGpa: System.out.print("Students GPA: ");
                        try
                        {
                            System.out.format("%.2f\n\n",gpaCalculator.calcGpa(courseData));
                        }
                        catch(IllegalArgumentException exc)
                        {
                            System.out.println(exc.getMessage());
                        }
                        break;

                case addCrs: userAddCourse(courseData, console);
                        break;

                case rmCrs: userRemoveCourse(courseData, console, courseSearcher);
                        break;

                case editCrs: userEditCourse(courseData, console, courseSearcher);
                        break;

                case strData: System.out.println("Storing database!");
                        userStore(courseData, console);
                        break;

                case reLdData: userReload(courseData, console);
                        break;

                case exit: System.out.println("Exiting");
                        System.exit(0); // exit successfully

                // Catch all (commands that are not 1-4,9)
                default: System.out.println("The command you entered is not recognized.\n");
                        break;
            }
        }
    }

    /**************************************************************
    * Method: userAddClass()                                      *
    * Purpose: add a user class to database                       *
    *                                                              *
    * Parameters: Database: Scanner: targetbase, console scanner *
    * Returns: void:                   N/A                         *
    ***************************************************************/
    private static void userAddCourse(Database targetDatabase, Scanner console)
    {
        System.out.println("Enter your class in the following format:\n"  +
                           "201003/CE-320/4/Microcomputers I/B+/N\n"      +
                           "yyyytt/corsnum/credit/title/grade/excluded\n" +
                           "yyyy is year tt is term (01,02,03,04)     "  );
        String userInput = ""; // User input string to be feed to addCourse
        Course tempCourse = new Course(); // course to be added
        String dateString = ""; // Temp string for separating the year and semester
        userInput = console.nextLine(); // get users input
        if(userInput == "")
        {
            System.out.println("You didn't enter anything");
            return;
        }
        Scanner pieces = new Scanner(userInput); // Scanner for spliting string


        // setTermTaken, and setExcludeFlag will throw a parse error
        //      if data sent is not in the correct format
        try
        {
            pieces.useDelimiter("/");
            dateString = pieces.next();
            if(dateString.length() != 6)
            {
                throw new ParseException("Year/Term is wrong length",0);
            }
            tempCourse.setYearTaken(dateString.substring(0, 4)); // Set year to string char's 0-4 (year)
            tempCourse.setTermTaken(dateString.substring(4, 6)); // Set term taken to the 2 digit semester code
            tempCourse.setCourseNumber(pieces.next());             // Set the course number
            tempCourse.setCreditCount(pieces.nextInt());         // Set the number of credits the class is worth
            tempCourse.setCourseTitle(pieces.next());             // Set the course title
            tempCourse.setCourseGrade(pieces.next().toUpperCase()); // Set the course grade
            tempCourse.setExcludeFlag(pieces.next());             // Set the exclude flag
        }
        catch(ParseException exc)
        {
            System.out.println(exc.getMessage() + " Your input is ignored!\n");
            return;
        }
        catch(InputMismatchException exc)
        {
            System.out.println(exc.getMessage()+"your input is ignored\n");
            return;
        }
        targetDatabase.addCourse(tempCourse);
        System.out.println("\n"); // extra space for prettiness
    }

    /**************************************************************
    * Method: userRemoveCourse()                                  *
    * Purpose: remove a user class to database                     *
    *                                                              *
    * Parameters: Database: Scanner: targetbase, console scanner *
    * Returns: void:                   N/A                         *
    ***************************************************************/
    private static void userRemoveCourse(Database targetDatabase, Scanner console,
            CourseSearch courseSearcher)
    {
        int promptCtr = 0; // Counter for seeing how many time the user was prompted
        System.out.println("Please enter the course number you would like to remove:");
        String userInput = console.next(); // users course input
        console.nextLine(); // Clear the scanner's buffer by going to the return char
                            //    still viable if a newline char is pasted into terminal
        LinkedList<Course> returnResults; // Results from the course search
        int deletedCounter = 0; // counter for number of courses deleted
        try
        {
            System.out.print("Course search, ");
            returnResults = courseSearcher.findByNumber(userInput, targetDatabase);
        }
        catch (NoSuchFieldException exc)
        {
            System.out.println("No results were found!\n");
            return;
        }
        System.out.println("Please enter the term you would like to remove it from " +
                           "(yyyytt): ");
        String termInput = console.next(); // capture the term to delete from
        console.nextLine(); // Clear the scanner's buffer by going to the return char
                            //     still viable if a newline char is pasted into terminal
        for(int index=0;index<returnResults.size();)
        {
            // Print the course as long as it matches year and term
            if(termInput.equalsIgnoreCase(returnResults.get(index).getTermTakenRaw()))
            {
                printCourse(returnResults, index);
                System.out.println("Would you like to delete(y/n):");
                userInput = console.next();
                if(userInput.equalsIgnoreCase("y"))
                {
                    for(int index2=0;index2<targetDatabase.getDatabaseSize();index2++)
                    {
                        if(targetDatabase.get(index2).getTerm().equals(
                                returnResults.get(index).getTermTakenRaw()))
                        {
                            // remove course from lower list
                            targetDatabase.remove(index2, returnResults.get(index));
                            deletedCounter++;
                            break; // exit for loop as term search is done
                        }
                    }
                }
                else{/* do nothing */}
                promptCtr++;
            }
            index += 2; // because of storage in results add 2 instead of 1
        }
        if(promptCtr>0)
            System.out.println("You deleted " + deletedCounter + " course(s)!\n");
        else
            System.out.println("There were no courses with the specified term!\n");
    }

    /************************************************************
    * Method: userRemoveCourse()                               *
    * Purpose: remove a user class to database                   *
    *                                                            *
    * Parameters: Database: Scanner: targetbase, console,        *
    * Returns: void:                   N/A                       *
    ************************************************************/
    private static void userEditCourse(Database targetDatabase, Scanner console, CourseSearch courseSearcher)
    {
        int promptCtr = 0; // Counter for seeing how many time the user was prompted
        System.out.println("Please enter the course number you would like to edit:");
        String userInput = console.next(); // users course input
        console.nextLine(); // Clear the scanner's buffer by going to the return char
                            //    still viable if a newline char is pasted into terminal
        LinkedList<Course> returnResults = null; // Results from the course search
        int editedCounter = 0; // For the number of edited courses
        try
        {
            System.out.print("Course search, ");
            returnResults = courseSearcher.findByNumber(userInput, targetDatabase);
        }
        catch (NoSuchFieldException exc)
        {
            System.out.println("No results were found!\n");
            return;
        }
        System.out.println("Please enter the term you would like to edit it in " +
                           "(yyyytt): ");
        String termInput = console.next(); // capture the term to delete from
        console.nextLine(); // Clear the scanner's buffer by going to the return char
                            //     still viable if a newline char is pasted into terminal

        for(int index=0;index<returnResults.size();)
        {
            // Print the course as long as it matches year and term
            if(termInput.equalsIgnoreCase(returnResults.get(index).getTermTakenRaw()))
            {
                printCourse(returnResults, index);
                System.out.println("Would you like to Edit?(y/n):");
                userInput = console.next();
                if(userInput.equalsIgnoreCase("y"))
                {
                    for(int index2=0;index2<targetDatabase.getDatabaseSize();index2++)
                    {
                        if(targetDatabase.get(index2).getTerm().equals(
                                returnResults.get(index).getTermTakenRaw()))
                        {
                            // remove course from lower list
                            targetDatabase.remove(index2, returnResults.get(index));
                            // create a new scanner as the old one causes problems
                            userAddCourse(targetDatabase, new Scanner(System.in));
                            editedCounter++;
                        }
                    }
                }
                else{/* do nothing */}
                promptCtr++;
            }
            index += 2; // because of storage in results add 2 instead of 1
        }
        if(promptCtr>0)
            System.out.println("You edited " + editedCounter + " course(s)!\n");
        else
            System.out.println("There were no courses with the specified term!\n");
    }

    /*************************************************************
    * Method: userStore()                                        *
    * Purpose: store database to file                           *
    *                                                            *
    * Parameters: Database: Scanner: targetbase, console scanner *
    * Returns: void:                    N/A                      *
    *************************************************************/
    private static void userStore(Database targetDatabase, Scanner console)
    {
        System.out.println("Enter a file name to save to: ");
        String userInput = console.next(); // users course input
        console.nextLine(); // Clear the scanner's buffer by going to the return char
                            //    still viable if a newline char is pasted into terminal
        try
        {
            targetDatabase.storeDatabase(userInput);
        }
        catch(IllegalArgumentException exc)
        {
            System.out.println("\nAre you trying to break me? There is nothing to store!\n");
        }
    }

    /*************************************************************
    * Method: userReload()                                       *
    * Purpose: reload a database from file                       *
    *                                                            *
    * Parameters: Database: Scanner: targetbase, console scanner *
    * Returns: void:                    N/A                      *
    *************************************************************/
    private static void userReload(Database targetDatabase, Scanner console)
    {
        System.out.println("Enter a file to load, within the local directory: ");
        String userInput = console.next(); // users course input
        console.nextLine(); // Clear the scanner's buffer by going to the return char
                            //    still viable if a newline char is pasted into terminal

        // Cast to string array to pass to load database
        String userInputArray[] = {userInput};
        targetDatabase.removeAll(); // destroy old data
        try
        {
            // attempt to load the database with file location
            //   given at runtime
            targetDatabase.loadDatabase(userInputArray);
        }
        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("\nDatabase reloaded!\n");
    }

    /************************************************************
    * Method: printCourse()                                       *
    * Purpose: print a singular course                           *
    *                                                           *
    * NOTE: this was making all my methods too long so...        *
    *                                                            *
    * Parameters: Database: LinkedList: Index:                   *
    *                targetDatabase, returnResults, index        *
    * Returns: void:                   N/A                       *
    ************************************************************/
    private static void printCourse(LinkedList<Course> returnResults, int index)
    {
        System.out.print(
            returnResults.get(index).getCourseNumber()     + ": " +
            returnResults.get(index).getCourseTitle()      + " (" +
            returnResults.get(index).getCreditCount()      + "). "+
            returnResults.get(index).getTermTaken()        + " "  +
            returnResults.get(index).getYearTaken()         + " "  +
            returnResults.get(index).getCourseGrade()      + "\n"    );
    }
}
```