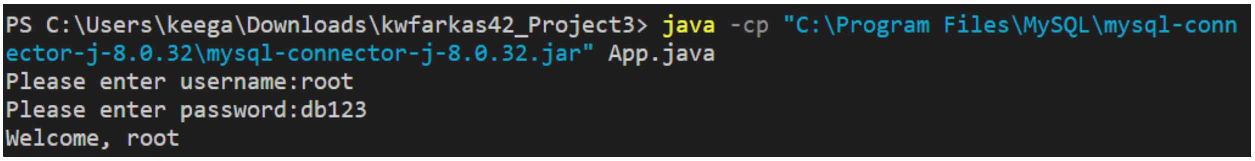
How to build and run the application

1. Install jdk from https://www.oracle.com/java/technologies/downloads/#jdk20-windows
2. Install a connector from https://dev.mysql.com/downloads/connector/j/
3. Make sure that your MYSQL server is running
4. In the directory that has ‘App.java’, run the following command to compile:
    - javac App.java
5. Then run the following command replacing <path to .jar> with the path to the .jar file that was downloaded in step 2 to run the application. My path is: C:\Program Files\MySQL\mysql-connector-j-8.0.32\mysql-connector-j-8.0.32.jar
    - java -cp “<path to .jar>” App.java

User Manual

Connecting to the server

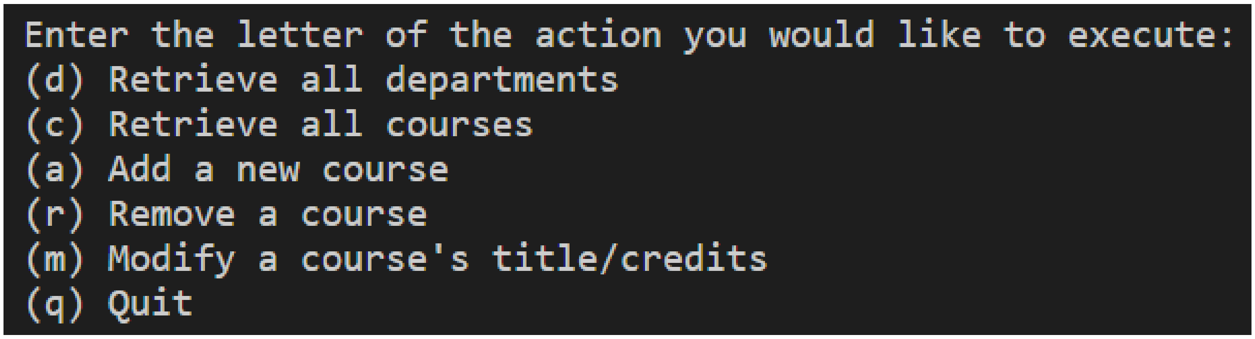
The application will first ask for a username and password. If the connection fails with the username and password you give, then it will ask if you want to try again or just quit the application. You will know that you have successfully connected when it says “Welcome, <username>”.



```
PS C:\Users\keega\Downloads\kwfarkas42_Project3> java -cp "C:\Program Files\MySQL\mysql-conn
ector-j-8.0.32\mysql-connector-j-8.0.32.jar" App.java
Please enter username:root
Please enter password:db123
Welcome, root
```

Main menu

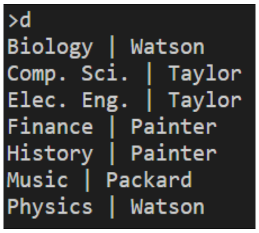
The main menu will give you 6 options. It will accept upper-case or lower-case, but if you enter anything besides one of the letters, it will tell you to try again.



```
Enter the letter of the action you would like to execute:
(d) Retrieve all departments
(c) Retrieve all courses
(a) Add a new course
(r) Remove a course
(m) Modify a course's title/credits
(q) Quit
```

(d) Retrieve all departments

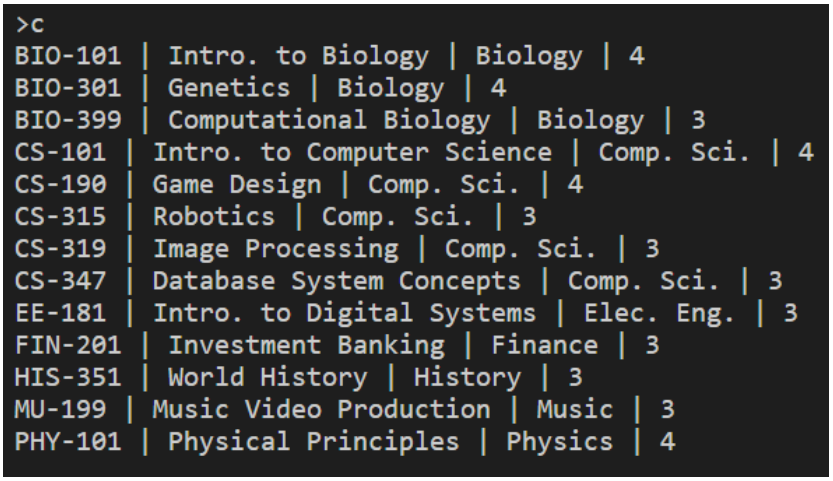
If you choose to retrieve all departments, nothing will be requested from you, but all of the departments will be displayed in “department | building” format.



```
>d
Biology | Watson
Comp. Sci. | Taylor
Elec. Eng. | Taylor
Finance | Painter
History | Painter
Music | Packard
Physics | Watson
```

(c) Retrieve all courses

If you choose to retrieve all courses, nothing will be requested from you, but all of the courses will be displayed in “course ID | title | department | credits” format



```
>c
BIO-101 | Intro. to Biology | Biology | 4
BIO-301 | Genetics | Biology | 4
BIO-399 | Computational Biology | Biology | 3
CS-101 | Intro. to Computer Science | Comp. Sci. | 4
CS-190 | Game Design | Comp. Sci. | 4
CS-315 | Robotics | Comp. Sci. | 3
CS-319 | Image Processing | Comp. Sci. | 3
CS-347 | Database System Concepts | Comp. Sci. | 3
EE-181 | Intro. to Digital Systems | Elec. Eng. | 3
FIN-201 | Investment Banking | Finance | 3
HIS-351 | World History | History | 3
MU-199 | Music Video Production | Music | 3
PHY-101 | Physical Principles | Physics | 4
```

(a) Add a new course

If you choose to add a new course, course ID, title, department, and credits will be requested from you. None of them can be empty. They have the following constraints:

Course ID:

- Must be 2-4 capital letters followed by a dash and 3 digits.(see first column of above image for examples)
- Must not already exist in the course table of the database

Title:

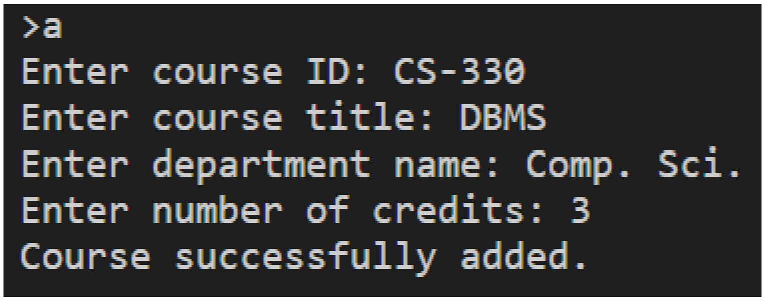
- Must be 50 characters or less

Department:

- Must be 20 characters or less
- Must already exist in the department table of the database

Credits:

- Must be an integer.(no decimals)
- Must be 2 digits or less
- Must be greater than 0

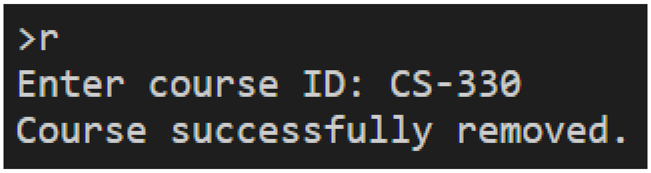


```
>a
Enter course ID: CS-330
Enter course title: DBMS
Enter department name: Comp. Sci.
Enter number of credits: 3
Course successfully added.
```

(r) Remove a course

If you choose to remove a course, course ID will be requested from you with the following constraints:

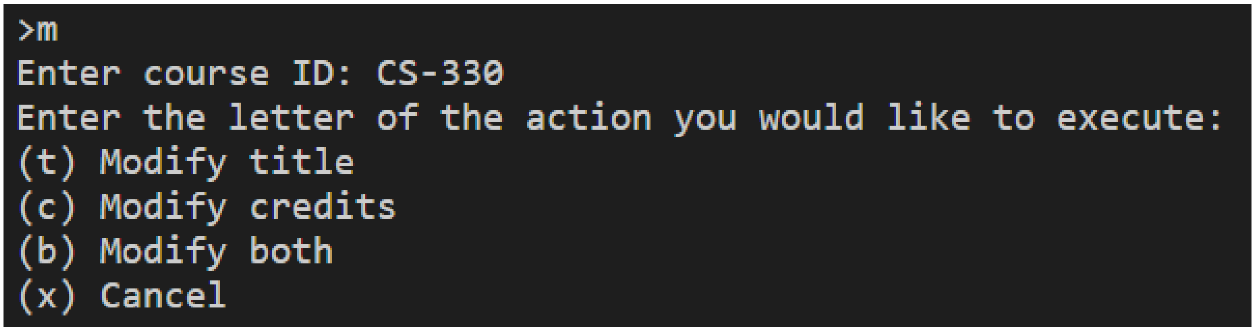
- Must be 2-4 capital letters followed by a dash and 3 digits.
- Must already exist in the course table of the database.



```
>r
Enter course ID: CS-330
Course successfully removed.
```

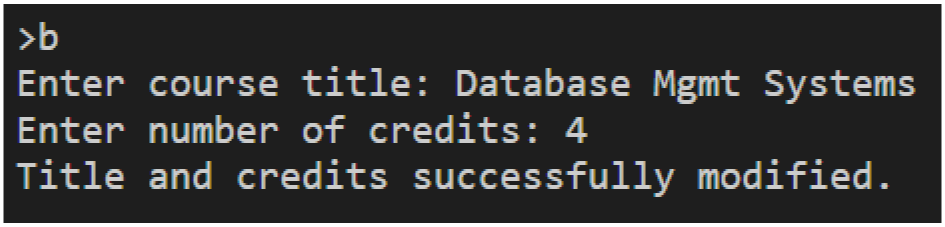
(m) Modify a course’s title/credits

If you choose to modify a course’s title/credits, course ID will be requested from you with the same constraints as removing a course. If you give a course ID that does not exist, then you will be sent back to the main menu. If you give a course that does exist, then you will be given 4 more options. These options work the same way as the main menu and will ask you to try again if you give an invalid input.



```
>m
Enter course ID: CS-330
Enter the letter of the action you would like to execute:
(t) Modify title
(c) Modify credits
(b) Modify both
(x) Cancel
```

If you choose to modify the title, the title will be requested from you with the same constraints as adding a course and the title of the course will be updated. If you choose to modify the credits, the number of credits will be requested from you with the same constraints as adding a course and the credits will be updated. If you choose to modify both, then both will be requested with the same constraints as adding a course and both will be updated. If you choose to cancel, then you will be sent back to the main menu.

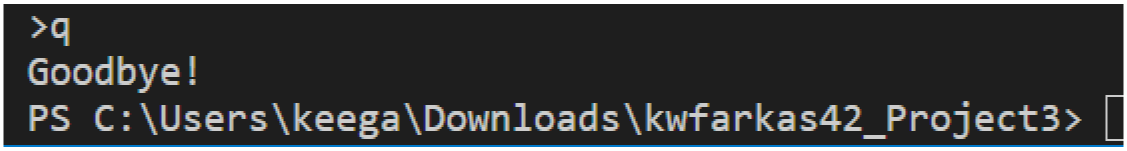


```
>b
Enter course title: Database Mgmt Systems
Enter number of credits: 4
Title and credits successfully modified.
```

\

(q) Quit

If you choose to quit, then the application will say “Goodbye!” and safely exit.



```
>q
Goodbye!
PS C:\Users\keega\Downloads\kwfarkas42_Project3>
```

Implemented Functionality

- Validation of user inputs
- Intuitive CLI
- Prepared statements when using user inputs in query, however not when there was no user input because it felt more efficient to use a normal statement
- Retrieve all info about departments, without info about their budgets
- Retrieve all info about courses(course_id, title, dept_name, and credits)
- Add a course
- Delete a course
- Update course’s title and/or number of credits it is worth

Assumptions

- None of the values can be empty(course_id, title, etc.)
- Course ID must be 2-4 capital letters followed by a dash and 3 digits
- Normal statements are fine when there is no user input being used