ENSF 409
Winter Semester 2021
Assignment 09

Assignment Instructions

Complete and submit exercise 20.3 (Lesson 20 Part 2). This assignment is due on March 22, 2021 by 5:59 PM Mountain Time.

General Instructions

Academic Integrity and Collaboration

This is an individual assignment. Your submission must be your own original work. You may not work with others to complete the assignment, although you may collaborate on exercises which are not submitted. It is not academic misconduct to ask for help on the discussion boards, but you may not copy code or complete answers from your peers.

Submission

You must submit your assignment in the specified format. Due to the number of students in the course, we are using automated grading. If your submission does not have the correct folder structure and file names, it will not be marked. This is a strict requirement.

File and directory names are case-sensitive.

You must submit your assignment to the appropriate D2L dropbox folder. You may submit multiple times before the assignment deadline, but only the last submission will be graded. Previous uploads are discarded by D2L. Therefore, each upload must be a complete submission. Do not submit multiple files across separate submissions.

The assignment must be submitted as a single zip folder named with your student ID. Furthermore, when the file is unzipped, the contents must be in a folder with the same name. You may wish to verify that your zip file has been correctly generated (includes the student ID directory, contains all files). You can do this by copying the zip file into another folder, unzip it, and examine the structure.

Within the folder with your ID number, you must create a subdirectory for each exercise within the assignment. Use lowercase and employ a _ to separate sub exercise numbers. For example, Exercise 1.3 should be in a folder `exercise1_3`.

Below is an example for submitting Assignment 1, assuming a student ID of *1234765*.

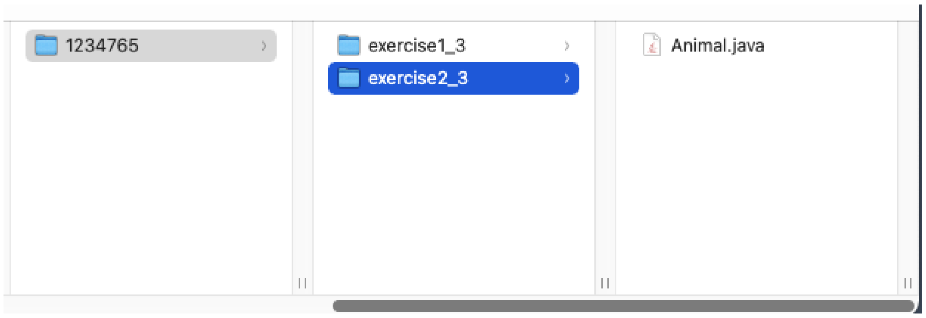
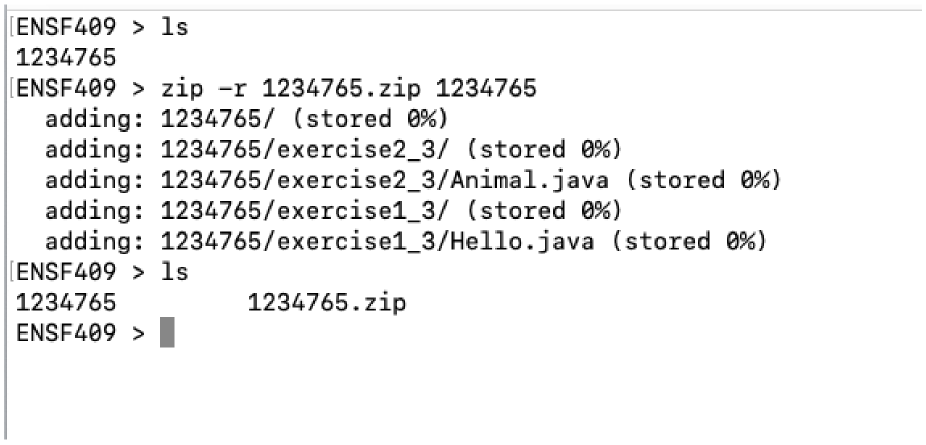
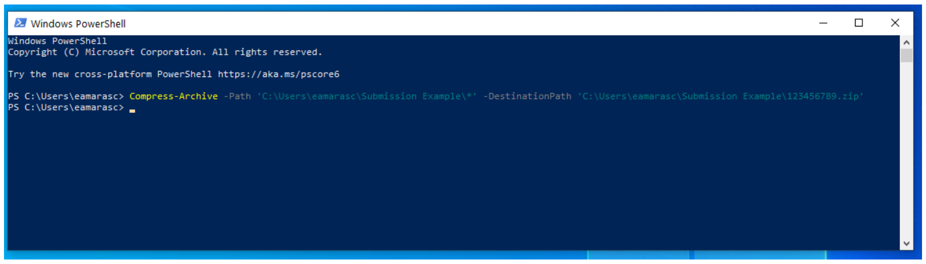


Figure 1: The folder structure for Assignment 1



```
ENSF409 > ls
1234765
ENSF409 > zip -r 1234765.zip 1234765
  adding: 1234765/ (stored 0%)
  adding: 1234765/exercise2_3/ (stored 0%)
  adding: 1234765/exercise2_3/Animal.java (stored 0%)
  adding: 1234765/exercise1_3/ (stored 0%)
  adding: 1234765/exercise1_3/Hello.java (stored 0%)
ENSF409 > ls
1234765         1234765.zip
ENSF409 >
```

Figure 2: Creating the zip file in Mac/Linux



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\eamarasc> Compress-Archive -Path 'C:\Users\eamarasc\Submission Example\*' -DestinationPath 'C:\Users\eamarasc\Submission Example\123456789.zip'
PS C:\Users\eamarasc>
```

Figure 3: Creating the zip file in Windows

Evaluation

Code which does not compile will not receive any points. Your code must compile and execute correctly to receive full marks. Assignments are graded using OpenJDK 11. While we do not anticipate that any of the assignments in this course will execute differently in any version of Java from 8 on, you should use this version for optimal results.

When style conventions are introduced in lessons, all subsequent submissions should adhere to the conventions.

Deadline

All homework assignments are due at 17:59 (5:59 PM) Mountain Time. You are responsible for the conversion between Mountain Time and your local time zone, if applicable. Be aware that the switch from Mountain Standard Time to Mountain Daylight Time occurs during the term.

It is recommended that you do not leave your submission until the last minute in case of technical issues. Once the dropbox folder closes, you will not be able to submit your assignment. Late submissions will not be accepted.

If there is a technical problem and you are unable to submit via the dropbox, do not email your assignment. The University of Calgary email system will remove zip files. Instead, you may email a link to download your .zip folder from OneDrive. The email must be received by Dr. Barcomb and Dr. Marasco before the assignment deadline and the OneDrive folder should not show any changes to the .zip file after the deadline has passed.

ENSF 409

Exercises - Lesson 20, Part 2

The following exercises are described in the video for Lesson 20, Part 2.

Exercise 20.2

1. Download and extract MySQL Connector/J on your computer
2. Follow along with the demonstration to practice database connections and manipulations

- Download and extract MySQL Connector/J
  - https://dev.mysql.com/downloads/connector/j/
  - MySQL Connector/J
  - May already be included with some Windows installations
  - Can use platform-independent version
  - Also uploaded to course repository
- Follow along with the tutorial demonstration
  - Code can be found in the course repository

Tip: Find the code examples in the course repository and follow along with the demonstration.

Exercise 20.3

1. Download the `competition.sql` file from the course repository
2. In the MySQL command line client, create the competition database using the command "source *full-path-to-competition-file*"
3. Create a program called Registration.java to access and update the provided database
4. An example main that provides some testing data is provided in the course repository

- Class must be named `Registration`
- Constructor should initialize three public final String data members called `DBURL, USERNAME, and PASSWORD`
- You must provide getters for each of these data members
- Provide the method `initializeConnection`
  - Creates a connection to the database
  - No arguments, does not return anything
- Provide the method `selectAllNames`
  - Takes in a String value representing a database table name
  - Returns a String containing any last and first names in the specified table
  - Each last/first name combination should be listed as LName, FName on its own line (no new line at the beginning or end)
- Provide the method `showStudios`
  - No arguments
  - Returns a String containing a list of all names in the Studio table
  - Each name should be listed on its own line (no new line at the beginning or end)
- Provide the method `insertNewCompetitor`
  - Takes in six arguments as they appear in the Competitor table
  - If the specified teacher is not included in the Teacher table, throw an `IllegalArgumentException`
  - If the competitor is not between the ages of 5 and 18, throw an `IllegalArgumentException`
  - If above conditions are met, insert the competitor into the Competitor table
  - No return value
- Provide the method `registerNewTeacher`
  - Takes in seven arguments as they appear in the Teacher and Studio tables
  - If the specified teacher is already registered, throw an `IllegalArgumentException`
  - If the teacher's studio is not listed in the Studio table, add the studio's information
  - Insert the teacher into the Teacher table
  - No return value
- Provide the method `deleteCompetitor`
  - Takes in a competitor's id as a String
  - Deletes the specified competitor from the Competitor table
  - No return value
- Provide the method `deleteTeacher`
  - Takes in a teacher's id as a String
  - Deletes the specified teacher from the Teacher table
  - No return value
- You may add any private data members and methods you wish
- All database-related try/catch statements should use the `SQLException`
- No required behavior for the catch statement (a print statement is fine)
- All resources must be released
- Method documentation must explain if/how each method is using the database