Name: Shilpa Shree

ID #: 1001540443

**READ ME: Instructions on how to execute code**

1. Install Eclipse Oxygen and MySQL Workbench.
2. Create a database in MySQL Workbench(Created test) and table (Shilpa)
3. Set the class path to mysql-connector-5.1.18.jar files.
4. Compile Main.java and DBMethods.java
5. Execute the code “java Main.java <Command Name> <Optional Parameter1> <Optional Parameter2>

**DETAILS ABOUT THE CODE:**

The code takes an argument from user to decide if the user wants to implement the add, list or delete functionality.

Depending upon the functionality the respective function (*Add(), List(), Delete\_PhoneName(), Delete\_PhoneNumber()* ) is called.

The add() function accepts input from the user and performs Input Validation before adding it to the database.

There are two additional methods checkname() and checkno(). These functions compare the inputs from the command line arguments to the respective regex string for Input Validation. Both the functions are called from add().(Delete function calls these two methods for validating and inspecting which type of input is it before performing delete operation)

The list() function lists the data from the database.

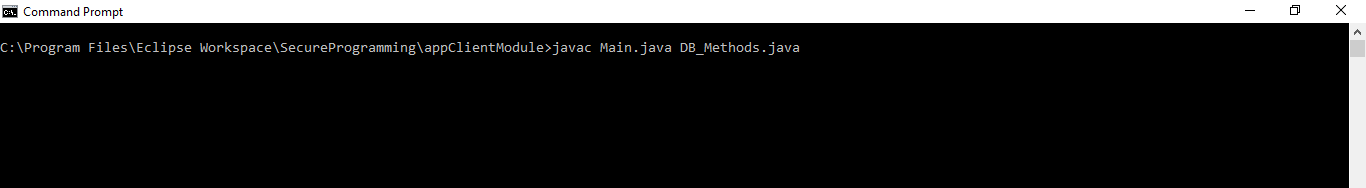
The delete() function accepts a key string from the command line arguments. It checks if it Phone\_Name or PhoneNumber using regex pattern and deletes the entire user record based on any one of these column name.

**COMPILATION INSTRUCTIONS:**

1. In cmd Prompt set the class path using

*set CLASSPATH=%CLASSPATH%; C:\Program Files\Java\jdk1.8.0\_144\jre\lib\ext\ mysql-connector-java-5.1.18-bin.jar;*

1. Set JAVA\_HOME AND JRE\_HOME environment variables. Set Path environment variables.
2. Compile the code using javac command as follows:



**ASSUMPTIONS MADE:**

**Phone Number:**

(\\+?\\d{1,3})?\\s{0,1}- This pattern matches with country code followed by optionally space.

\\(?[0-9]{0,3}\\)?)\\s{0,1}-This pattern matches with the state code followed by optionally space.

\\d{3}(\\s{1}|\\-)\\d{4})- Matches the valid phone number.

(\\d{5}|\\d{5}[.]\\d{5})- To match the phone number of type 12345 and 12345.12345.

**Name:**

([a-zA-Z]['])?- To match optional descendant.

[a-zA-Z]+[/,]?\\s\*- To match First Name.

[a-zA-Z]\*\\.?- Optional last name with Optional ‘.’

([a-zA-Z]['])?[a-zA-Z]+[/-]?[a-zA-Z]\*\\s\*- Middle name with optional descendant and optional ‘-‘

**PROS/CONS:**

Pros –

The Regex used to validate input has been developed based on White-listing which ensures that the program will allow only valid data to be added to the file.

Input validation is performed on every input given to the code.

Since regex has been used and validation is performed at every point of data entry in the code, there are less chances of Injection.

Constructed Parameterized SQL statements using bind parameters to prevent SQL injection

Cons -

The approach of this code does not perform a check on the length of the input given. Hence a very large string may lead to buffer overflow. However, since Java usually prevents the occurrence of Buffer Overflow, it is assumed that the code will work fine.

The Regex used to validate Input, is developed based on the valid and invalid strings, however, there will be certain scenarios which have not been considered.

**REFERENCES:**

<https://www.tutorialspoint.com/java/java_regular_expressions.htm>

<http://cs.lmu.edu/~ray/notes/regex/>

<http://stackoverflow.com/questions/756567/regular-expression-for-excluding-special-characters>

<https://www.google.com/search?rlz=1C1CHZL_enUS760US760&q=o%27+prefix&oq=why+do+you+use+o%27+in+names&gs_l=psy-ab.3.2.0i71k1l4.0.0.0.10153.0.0.0.0.0.0.0.0..0.0....0...1..64.psy-ab..0.0.0....0.AWczTWvRXEY>

<https://stackoverflow.com/questions/20324476/cant-connect-to-my-sql-database>

<https://stackoverflow.com/questions/43411184/com-mysql-jdbc-exceptions-jdbc4-mysqlsyntaxerrorexception-unknown-database-xyz>

<https://www.mkyong.com/jdbc/how-to-connect-to-mysql-with-jdbc-driver-java/>

<http://www.vogella.com/tutorials/MySQLJava/article.html#jdbc>

<https://dev.mysql.com/downloads/file/?id=471660>

<https://coderanch.com/t/303459/databases/closing-Connection-returning-ResultSet>

<https://coderwall.com/p/609ppa/printing-the-result-of-resultset>