

## CS3354 – Spring 2019 – Assignment 2

Due date: Friday, Feb. 22 at 11:55 p.m.

**Project Title:** Sentiment Analysis (stage 2)

**Goal:** The goal of this assignment is to help students familiarize themselves with the following Java programming concepts:

- Object Oriented Design and Inheritance
- Java Exception Handling
- Java Object Serialization
- Using existing codebase and libraries

### Description:

In your first assignment, you created a program that could classify movie reviews based on their sentiment. In this assignment we will extend the functionality of the previous assignment in a number of ways:

1. We will use an existing machine learning library to classify movie reviews as opposed to counting positive and negative words. The hope is that we will be able to achieve a better classification accuracy using a more advanced tool.
2. We will add some simple functionality for maintaining a database of movie reviews, which can be extended by adding new movie reviews. A user can later search that database to refer back to specific movie reviews.
3. We will be able to save the database and load it again the next time the program runs.

Your program should present the user with the following menu:

```
0. Exit program.
1. Load new movie review collection (given a folder or a file path).
2. Delete movie review from database (given its id).
3. Search movie reviews in database by id or by matching a substring.
```

### Notes:

- The above menu should be coded in a loop, so that the user can choose among the different options multiple times, until they choose option '0', in which case the program terminates.
- Every time your program loads, it should first check if there exists a **database file** (serialized object) in its working directory. If such a file exists, it should load its contents (movie reviews) into the main memory (a HashMap can be used). When the program exits (user selects action '0'), it should save the new database contents back to the database file, replacing the old one.

- When the user selects option “1”:
  - The program should also ask the user to provide the real class of the review collection (if known). The user can choose from the options: 0. Negative, 1. Positive, and 2. Unknown.
  - Upon loading each review, your program should assign a unique ID to each review, which should not conflict with existing ones, and it should also assign the value of the real class (as provided by the user).
  - Then the program should automatically classify each review, using the external library “`sentiment.jar`” and assign a value to the “`predictedClass`” field of each review. The overall classification accuracy should also be reported, if the real class is known.
  - Finally, the newly loaded reviews should be added to the permanent database.
- When the user selects option “3”, the results should be printed in a formatted manner. The printed information should be a table with each row showing: review ID, first 50 characters of review text, predicted class, real class.

#### Tasks:

1. Extend the given code to create an object-oriented Java program that implements the functionality described above. Make use of the inheritance and polymorphism concepts that we saw in class.
2. Try to make your program as robust as possible, by using Exception handling to deal with possible problems that may occur during the program execution.
3. Save your database as a **serialized object**.
4. Use javadoc comments for all of your classes and methods.

#### Logistics:

- This assignment should be completed by **individually or in pairs**.
- The provided code should be cloned from <http://git.txstate.edu/> according to the instructions provided in class.
  - The completed code should be pushed back to Git class repository.
    - Individual submission, for pairs – duplicate the code.
- You may use an IDE (BlueJ, Netbeans, etc) or just an editor and command line operations (`javac`, `java`) in Unix or Windows/DOS to develop your program.
- Use good design (don’t put everything in one class).
- Use a package for your classes and put your files in the appropriate directory structure.
- You don't need to create any GUI for this assignment. Command line operations are enough.
- Use a standard Java coding style to improve your program’s visual appearance and make it more readable. I suggest the BlueJ coding style: <http://www.bluej.org/objects-first/styleguide.html>