Carnegie Mellon University
Master of
Software Engineering

17-615: Data Structures and Algorithms
Assignment A6: Twitter

Learning Objectives: The objectives of this assignment is to gain practice considering which data structures to choose to solve a given problem.

Assignment Description: Design a simplified version of Twitter where users can post tweets, follow/unfollow another user and is able to see the 10 most recent tweets in the user's news feed. Your design should support the following methods:

1. **createUser(userID):** Create a user
2. **postTweet(userId, tweetId)**: Compose a new tweet.
3. **getNewsFeed(userId)**: Retrieve the 10 most recent tweet ids in the user's news feed. Each item in the news feed must be posted by users who the user followed or by the user herself. Tweets must be ordered from most recent to least recent.
4. **follow(followerId, followeeId)**: Follower follows a followee.
5. **unfollow(followerId, followeeId)**: Follower unfollows a followee.

I/O Requirements:

Use a CLI to take input for example:

1. Post a tweet
2. Get News Feed
3. Create a User
4. Follow
5. Unfollow

Further Instructions:

- When posting a tweet the CLI should ask for tweetId and userId and print a success/error message (In the case the user does not exist/ tweetId already exists)
- When getting News Feed the CLI should ask for the userId and print a list of top 10 tweetId ordered from most recent to least with a success message and an error message when user does not exist
- createUser(userId), should give appropriate success/error message
- For follow function the CLI should prompt for userid1 and userid2 and print a success/error message
- For unfollow function the CLI should prompt for userid1 and userid2 and print a success/error message

Example:

```
Twitter twitter = new Twitter();

// User 1 posts a new tweet (id = 5).
twitter.postTweet(1, 5);

// User 1's news feed should return a list with 1 tweet id -> [5].
twitter.getNewsFeed(1);

// User 1 follows user 2.
twitter.follow(1, 2);

// User 2 posts a new tweet (id = 6).
twitter.postTweet(2, 6);

// User 1's news feed should return a list with 2 tweet ids -> [6, 5].
// Tweet id 6 should precede tweet id 5 because it is posted after tweet id 5.
twitter.getNewsFeed(1);

// User 1 unfollows user 2.
twitter.unfollow(1, 2);

// User 1's news feed should return a list with 1 tweet id -> [5],
// since user 1 is no longer following user 2.
```

```
twitter.getNewsFeed(1);
```

Important: You need to provide detailed execution instructions that will work on any system. Your program should run on any computer with a JVM without any additional software required. Be careful about using IDEs and building in external dependencies. We will not debug your software or change the configuration of our system to test your software. If it doesn't run you won't get credit. **Test your software and execution instructions on a fresh machine prior to submission**

Remember to provide well commented code that handles erroneous input.

Submission Instructions: You will upload your source code and execution instructions to Canvas

Assignment Grading: Your grade will be calculated as follows:

- Code compiles: 30 pts
- Functionality executes as expected: 20 pts
- Conforms to guidelines given in the instructions: 20 pts
- Handles unexpected input: 15 pts
- Code is well written and documented: 15 pts