# Programming Assignment – 5
## CS 2133 Computer Science II
## *Due: 11/15/2022 at 11:59 PM on CANVAS*
## *50 points*

**Analyze the time complexity of the below code/programs and explain your answer briefly. (2.5 * 4 = 10 points)**

1. What is the time complexity of following code:

```
int  a = 0;
for  (i = 0; i < N; i++) {
    for  (j = N; j > i; j--) {
        a = a + i + j;
    }
}
```

2. What is the time complexity of following code:

```
int  a = 0, i = N;
while  (i > 0) {
    a += i;
    i /= 2;
}
```

3. Two loops in a row:

```
for (i = 0; i < A; i++) {
    sequence of statements
}
for (j = 0; j < B; j++) {
    sequence of statements
}
```

How would the complexity change if the second loop went to A instead of B?

4. What is the time complexity of the following code?

```
int n = 1000;
System.out.println("Your input is: " + n);
```

## Simple Sentiment Analysis (40 points)

Sentiment Analysis is a Big Data problem used to get the attitude or sentiment of the writer based on their written text. For instance, when analyzing movie reviews, "The film was a breath of fresh air" has a positive statement while "It made me want to poke out my eye balls" is negative.

One algorithm that we can use for this is to assign a numeric value to any given word based on how positive or negative that word is and then score the statement based on the values of the words. But, how do we come up with our word scores in the first place?

That's the problem that we'll solve in this assignment. In the attachments, you will see a data file *movieReviews.tsv* which contains movie reviews from the Rotten Tomatoes website which have both a numeric score as well as text. The data file has four columns – phraseID, sentenceID, the phrase and a sentiment score.

Your goal is to write a JAVA program that reads the given data file and aggregate the review score *per sentence*. You will need to create an instance of a class named *ReviewEntry* for each word in the file and track the total score assigned to the sentence.
You will need to write a function that computes the average score for each sentence and assigns the following meaning to a word based on the average score:

0-2: negative
2-3: neutral
3-4: positive

**[Part 1]** Create a class called *ReviewEntry* which has the following elements: (i) a variable to track the sentence ID, (ii) a variable to track the *total score* (which is the sum of all scores assigned to the sentence) and (iii) a variable to track the *number of phrases* in the sentence. Create an array of object instances of this class to account for each sentence in the file. You will have a total 8544 elements in the array.

**[Part 2]** Use FileIO operations in JAVA to read each entry in the *movieReviews.tsv* file and update the corresponding entry for the sentence. For example, you should update the object for sentenceID 1 with each corresponding entry in *movieReviews.tsv*

**[Part 3]** Once you finish loading the entries in the file, ask the user for a sentence ID as input and print the statistics of the corresponding sentence in the following format:

*Sentence ID 1 has 15 phrases with an average rating of 3.5. The overall sentiment is positive.*