CMPT435 Assignment1

Brian Sprague

September 15, 2020

1 Introduction

This is my LaTeX submission for Assignment 1. In it will be snippets of my code along with explanations on some of the parts. There are three classes in my assignment, including Assignment1, which is the main class, stack, and queue.

2 Code

```
import java.io.*;
public class Assignment1 {


public static void main(String[] args) throws Exception{
        String[] list = new String[666];
        Stack myStack = new Stack();
        Queue myQueue = new Queue();
        File file = new File("C:\\Users\\sprag\\OneDrive\\Documents\\downloads
            \\magicitems.txt");

    BufferedReader br = new BufferedReader(new FileReader(file));

        String st;
        while ((st = br.readLine()) != null) {
        int k=0;
    list[k] = st;
    k++;
        System.out.println(st);
        }


    int compare = 0;
    for(int i=0; i<666; i++) {
            for(int j = 0; j<list[i].length();j++) {
```

```
                        list[i].toLowerCase();
                        list[i].trim();
                        myQueue.enqueue(list[i]);
                        myStack.push(list[i]);
                        String queueTemp = myQueue.dequeue();
                        String stackTemp = myStack.pop();
                        compare = queueTemp.compareTo(stackTemp);
                        if(compare != 0) {
                          break;
                        }
                        if(compare == 0) {
                        System.out.println(list[i] + " is a palindrome.");
                        }

                }

        }


}

}
```

In this code, I read the magic items list into my code through the File command, which is then printed out in the while loop for proof of input. Then, the double for loop is used to both go through each word and then each letter of every inputted item. Then inside the for loop is where the palindrome comparison runs.

```
public class Stack{
        private class Node{
                String data;
                Node link;
        }
        Node top;
        public static boolean emptyList;

        public boolean isEmpty() {
                return top == null;
        }
        public void push(String x) {
                Node temp = new Node();
                if(temp == null) {
                        System.out.println("Overflow");
                }
                temp.data = x;
                temp.link = top;
```

```
                top = temp;
        }
        public String pop() {
                if(top == null) {
                        System.out.println("Underflow");
                }
                top = (top).link;
                return top.data;

        }

}
```

My Stack class includes the main stack functions such as push, pop, and isEmpty.

```
import java.util.*;
public class Queue {
    private ArrayList<String> list = new ArrayList<String>();

    public boolean isEmpty() {
        return list.size() == 0;
    }

    public void enqueue(String item) {
        list.add(item);
    }

    public String dequeue() {
        String item = list.get(1);
        list.remove(0);
        return item;
    }

    public String peek() {
        return list.get(0);
    }



}
```

My Queue class contains the main functions such as queue, dequeue and isEmpty.