Name: Brendan Yong **Tutorial Group ID**: W10	

Code

```
import java.io.*;
import java.util.Scanner;

public class textBuddy {

	private static Scanner scanner = new Scanner(System.in);

	private static BufferedReader file = null;

	private static String fileName = "";

	private static boolean exitMarker = false;

	private static void showToUser(String text) {
		System.out.println(text);
	}

	private static String executeCommand(String userCommand) throws Exception{
		String commandType = getFirstWord(userCommand);
		if (commandType == null){
			throw new Error("No command entered. Please enter command.");
		}

		if (commandType.equalsIgnoreCase("add")){
			String toAdd = removeFirstWord(userCommand);
			return add(toAdd);
		}

		else if (commandType.equalsIgnoreCase("delete")){
			String toDelete = removeFirstWord(userCommand);
			return delete(toDelete);
		}

		else if (commandType.equalsIgnoreCase("clear")){
			return clear();
		}

		else if (commandType.equalsIgnoreCase("display")){
			return display();
		}

		else if (commandType.equalsIgnoreCase("exit")){
			return exit();
		}
```

```
		else {
			return "Invalid command \""+ userCommand + "\" entered! Please enter
again.";
		}

	}

	private static String removeFirstWord(String userCommand) {
		return userCommand.replace(getFirstWord(userCommand), "").trim();
	}

	private static String getFirstWord(String userCommand) {
		String commandTypeString = userCommand.trim().split("\\s+")[0];
		return commandTypeString;
	}

	public static String add(String toAdd) throws Exception {
		file = new BufferedReader(new FileReader(fileName));
		String newText = "";
		String line = null;
		int num = 1;

		while((line = file.readLine()) != null){
			num = Integer.parseInt(line.split("\\.")[0]) + 1;
			newText = newText + line + "\n";
		}

		newText = newText + num + ". " + toAdd;
		BufferedWriter bw = new BufferedWriter(new FileWriter(fileName));
		bw.write(newText);
		bw.close();

		return "\"" + toAdd + "\" has been added to " + fileName + ".";
	}

	public static String delete(String toDelete) throws Exception {
		file = new BufferedReader(new FileReader(fileName));
		int num = checkNum(toDelete);

		if(num == 0){
			return "\"" + toDelete + "\" is not a number, or is in the wrong format!";
		}

		else {
			String newText = "";
			String deletedText = "";
			String line = null;
			while((line = file.readLine()) != null){

				if(!line.split("\\s+")[0].equals(num+".")){
					newText = newText + line + "\n";
				}
```

```
                        else {
                                deletedText = removeFirstWord(line);
                        }
                }
                BufferedWriter bw = new BufferedWriter(new FileWriter(fileName));
                bw.write(newText);
                bw.close();

                if (deletedText != ""){
                        return "\"" + deletedText + "\" has been deleted from " + fileName +
"."; 
                }

                else {
                        return "There is no corresponding text to delete in " + fileName + ".";
                }
        }
    }

    private static int checkNum(String param){
        try{
                int num = Integer.parseInt(param);
                return num;
        }
        catch (NumberFormatException e){
                return 0;
        }
    }

    public static String clear() throws Exception{
        file = new BufferedReader(new FileReader(fileName));
        BufferedWriter bw = new BufferedWriter(new FileWriter(fileName));
        bw.write("");
        bw.close();
        return "All content has been deleted from " + fileName + ".";
    }

    public static String display() throws Exception {
        file = new BufferedReader(new FileReader(fileName));
        String line = null;
        String allText = "";
        while((line = file.readLine()) != null){
                allText = allText + line + "\n";
        }

        if(allText == ""){
                return "There is no text in " + fileName + ".";
        }

        else {
                return allText.trim();
        }
    }
```

```
	private static String exit(){
		exitMarker = true;
		return "Thank you for using TextBuddy :)";
	}

	public static void main(String args[])throws Exception {
		try {
			fileName = args[0];
			showToUser("Welcome to TextBuddy. " + fileName + " is ready for use.");

			while(exitMarker == false){
				showToUser("Enter Command: ");
				String command = scanner.nextLine();
				String userCommand = command;
				String feedback = executeCommand(userCommand);
				showToUser(feedback);
			}
		}

		catch(Exception e){
			e.printStackTrace();
		}
	}
}
```

TestInput.txt

asdfghj
add fox
display
add cow

```
display
add baby
display
delete 1
delete 2
add another cow
display
clear
display
add

add 54678
delete 2
delete r
clear
exit
```

ExpectedOutput.txt

```
Welcome to TextBuddy. myText.txt is ready for use.
Enter Command:
Invalid command "asdfghj" entered! Please enter again.
Enter Command:
"fox" has been added to myText.txt.
Enter Command:
1. fox
Enter Command:
"cow" has been added to myText.txt.
Enter Command:
1. fox
2. cow
Enter Command:
"baby" has been added to myText.txt.
Enter Command:
1. fox
2. cow
3. baby
Enter Command:
"fox" has been deleted from myText.txt.
Enter Command:
"cow" has been deleted from myText.txt.
Enter Command:
"another cow" has been added to myText.txt.
Enter Command:
```

```
3. baby
4. another cow
Enter Command:
All content has been deleted from myText.txt.
Enter Command:
There is no text in myText.txt.
Enter Command:
"" has been added to myText.txt.
Enter Command:
Invalid command "" entered! Please enter again.
Enter Command:
"54678" has been added to myText.txt.
Enter Command:
"54678" has been deleted from myText.txt.
Enter Command:
"r" is not a number, or is in the wrong format!
Enter Command:
All content has been deleted from myText.txt.
Enter Command:
Thank you for using TextBuddy :)
```