# Professional Code Writing Lab

Download the following Project or StudentList.java and students.txt data  file from the following link
**`https://github.com/m-r-kushal/wpcl-test.git`**

if you have ssh configured

```
git clone git@github.com:m-r-kushal/wpcl-test.git
```

if you don't have ssh configured, use https instead

```
git clone https://github.com/m-r-kushal/wpcl-test.git
```

or if you don't have internet connection, please write the following code.

```
//File Name EmployeeManager.java
import java.io.*;
import java.util.*;

public class EmployeeManager {
    public static void main(String[] args) {
        // Check arguments
        if (args[0].equals("l")) {
            System.out.println("Loading data ...");
            try {
                BufferedReader r = new BufferedReader(
                        new InputStreamReader(
                                new
FileInputStream("employees.txt")));
                String l = r.readLine();
                String e[] = l.split(",");
                for (String emp : e) {
                    System.out.println(emp);
                }
            } catch (Exception e) {}
            System.out.println("Data Loaded.");
        } else if (args[0].equals("s")) {
            System.out.println("Loading data ...");
            try {
                BufferedReader r = new BufferedReader(
                        new InputStreamReader(
                                new
FileInputStream("employees.txt")));
                String l = r.readLine();
                System.out.println(l);
                String e[] = l.split(",");
                Random rand = new Random();
                int idx = rand.nextInt(e.length);
                System.out.println(e[idx]);
            } catch (Exception e) {}
            System.out.println("Data Loaded.");
        } else if (args[0].contains("+")) {
            System.out.println("Loading data ...");
            try {
                BufferedWriter w = new BufferedWriter(
                        new FileWriter("employees.txt", true));
                String n = args[0].substring(1);
                w.write(", " + n);
                w.close();
            } catch (Exception e) {}
            System.out.println("Data Loaded.");
        }
         else if (args[0].contains("?")) {
            System.out.println("Loading data ...");
            try {
                BufferedReader r = new BufferedReader(
                        new InputStreamReader(
                                new
FileInputStream("employees.txt")));
                String l = r.readLine();
                String e[] = l.split(",");
                boolean found = false;
                String s = args[0].substring(1);
                for (int i = 0; i < e.length && !found; i++) {
                    if (e[i].equals(s)) {
                        System.out.println("Employee found!");
                        found = true;
                    }
                }
            } catch (Exception e) {}
            System.out.println("Data Loaded.");
        } else if (args[0].contains("c")) {
            System.out.println("Loading data ...");
            try {
                BufferedReader r = new BufferedReader(
                        new InputStreamReader(
                                new
FileInputStream("employees.txt")));
                String l = r.readLine();
                char[] chars = l.toCharArray();
                boolean inWord = false;
                int count = 0;
                for (char c : chars) {
                    if (c == ' ') {
                        if (!inWord) {
                            count++;
                            inWord = true;
                        } else {
                            inWord = false;
                        }
                    }
                }
                System.out.println(count + " word(s) found " +
chars.length);
            } catch (Exception e) {}
            System.out.println("Data Loaded.");
        } else if (args[0].contains("u")) {
            System.out.println("Loading data ...");
            try {
                BufferedReader r = new BufferedReader(
                        new InputStreamReader(
                                        new
FileInputStream("employees.txt")));
                String l = r.readLine();
                String e[] = l.split(",");
                String n = args[0].substring(1);
                for (int i = 0; i < e.length; i++) {
                    if (e[i].equals(n)) {
                        e[i] = "Updated";
                    }
                }
                BufferedWriter w = new BufferedWriter(
                        new FileWriter("employees.txt"));
                w.write(String.join(",", e));
                w.close();
            } catch (Exception e) {}
            System.out.println("Data Updated.");
        } else if (args[0].contains("d")) {
            System.out.println("Loading data ...");
            try {
                BufferedReader r = new BufferedReader(
                        new InputStreamReader(
                                new
FileInputStream("employees.txt")));
                String l = r.readLine();
                String e[] = l.split(",");
                String n = args[0].substring(1);
                List<String> list = new
ArrayList<>(Arrays.asList(e));
                list.remove(n);
                BufferedWriter w = new BufferedWriter(
                        new FileWriter("employees.txt"));
                w.write(String.join(",", list));
                w.close();
            } catch (Exception e) {}
            System.out.println("Data Deleted.");
        }
    }
}
```

**Data File name**

`employees.txt`

**File Contents (Initial Stage)**

```
John Doe,Jane Smith,Alice Johnson,Bob Brown
```

Data file and java file should be in same location.

**Task #0** (Check the output by running the program with the command line argument like `l, s, +, ?, c, u`)

You should create git branch for every task with the name of the task, work on that task, commit the code with appropriate commit message and finally merge your branch to the master. For better understanding, write the commit message same as the task on each task. And check every time you change something, you didn't break anything.

Task #1. Update **Code Style for Better Consistency:**

- Ensure consistent indentation and spacing.
- Use proper braces and line brakes.

Task #2. **Application Now** Terminates **Early if the Number of Arguments Passed into It Is Wrong, Fix It:**

- Add a check at the beginning to ensure the correct number of arguments.

Task #3. **Makes** Improvements **to Variable Names:**

- Use meaningful variable names instead of single letters.

Task #4. **Refactors** Duplicate **File Read and Write Logic into Methods:**

- Create methods for reading and writing to the file to avoid code duplication.

Task #5. **Replaces String Literals with Constants, Storing Those Constants in a New Class Called Constants.java:**

- Define constants for file paths and date formats in a separate **Constants.java** file.

Task #6. **Remove Temporary Variables:**

- Eliminate unnecessary temporary variables.

Task #7. **Eliminates the 'done' Control-Flow Variable. Adds Better Response for Search Operation:**

- Use a more straightforward approach for the search operation.

Task #8. **Simplifies the Logic Behind the Count Operation:**

- Simplify the logic for counting words.

Task #9. **Adds Handling for Case When User Enters Invalid Arguments:**

- Add error handling for invalid arguments.

Task #10. **Add More Comments and Makes More Naming Improvements:**

- Add comments to explain the purpose of each section of the code.
- Improve variable and method names for better readability.