

## CSED232 Object Oriented Programming (Spring 2019)

### Programming Assignment #2

#### Voting System 구현

Due Date: 2019년 3월 29일 금요일 23:59

담당조교: 김태욱 (taewook101@postech.ac.kr)

## 1. Objective

본 과제에서는 회원들의 의견을 반영하여 의사결정을 내릴 수 있는 투표시스템을 구현해 보고, 이를 통해 Class 개념과 Design 방법을 익힌다. 메뉴 파싱 코드, 파일 입출력과 관련된 코드는 미리 제공 된다.

## 2. 과제 설명

### 2.1 시스템 개요

시스템 상에서 사용되는 텍스트 기반 메뉴 형태는 다음과 같다.

7

```
=====
num:1
=====
1. Register as a Member
2. Unsubscribe from System
3. Login
4. Logout
5. Delete Existing Vote Item
6. Add a New Vote Item
7. List All Vote Items
8. Cast a Vote
9. Program Exit
=====
```

Figure 1 텍스트 기반 메뉴 형태

본 시스템의 사용자는 1번 ~ 9번까지의 메뉴 중 하나의 메뉴를 선택하여 선택된 기능을 수행하게 된다. 시스템 상에는 크게 회원 관리 관련 기능, 투표 관리 기능이 있다.

#### 1) 회원 관리 기능

회원 관리 기능에는 회원 가입, 회원 탈퇴, 로그인, 로그 아웃이 있으며, 메뉴 선택 시 각각의 기능 수행에 필요한 정보를 사용자로부터 입력 받는다. 입력 받은 정보는 필요 시

시스템 상에 저장되며, 저장된 정보를 활용하여, 회원 중복 가입 여부 등과 같은 각 메뉴 수행에 필요한 verification과정이 수행된다. 회원 탈퇴 후, 동일한 주민번호로 재가입하는 것은 허용한다.

## 2) 투표 관리 기능

관련 기능에는 **투표 삭제**, **투표 제안**, **투표 리스트 조회**, **투표하기**가 있다. 투표 관련 정보로는, **투표 주제**, **투표별 항목**, **투표 항목별 득표 현황**이 있고, 구현된 시스템을 통해 이에 대한 관리가 이루어진다. 투표 주제는 시스템 상에 설정 가능한 최대 투표 주제 길이를 넘지 않는 문자열 정보이고, 투표 항목은 별도의 명칭이 없으며 각 항목별 득표 현황만이 관리 된다. 따라서, 사용자는 제안하고자 하는 투표의 주제와, 투표 항목 수만 입력하면, 투표 제안이 완료된다. 예를 들어, 사용자에 의해 투표 주제가 "vote 1" 이고, 투표 항목 수가 4인 투표를 제안하게 되면 시스템 상에는 다음과 같은 형태의 투표가 존재 할 수 있다.

**Vote Subject: vote1**

**Vote Counts per Item - item1: 0 item2: 0 item3: 0 item4: 0**

투표 제안은 회원 가입된 회원 누구나 가능하며, 각 회원은 투표 주제가 겹치지 않는 복수개의 투표를 제안할 수 있다. 모든 회원은 각 투표에 대해 하나의 투표권만을 갖는다. 투표 삭제, 투표 리스트 조회 또한 가입된 회원 누구나 가능하다.

## 2.2 메뉴 기능 설명

- 해당 메뉴가 선택될 시, 현재 선택된 메뉴 명을 출력하고(ex. : 1. Register as a Member), 다음 예시와 같이 **4.2 로그파일 형식**에 명시된 순서대로 각각의 정보 입력 요청 문장을 출력하고, 정보를 입력 받아 필요한 기능을 수행한다.
- **메뉴 선택은 9. Program Exit이 선택되기 전까지 반복 수행되며, 매 출력시 마다, "Num: 1" 과 같이 출력 횟수를 카운트하여 출력한다.**

```
=====
Num: 1
=====
1. Register as a Member
2. Unsubscribe from System
3. Login
4. Logout
5. Delete Existing Vote Item
6. Add a New Vote Item
7. List All Vote Items
8. Cast a Vote
```

```

9. Program Exit
=====
Select Menu:1
Selected Menu: 1. Register as a Member
1. Resident Registration Number: 1111
ID: id1
Password: pwd1
Registration Done!
-

```

- 메뉴 기능을 제대로 수행했을 경우, 화면상에 성공하였다는 메시지를 출력하며, 실패 시 실패하였다는 메시지를 출력하도록 한다.
- 또한 선택된 메뉴 번호 및 메뉴 선택 후 입력되는 정보는 commandLog.txt 파일에 기록한다.
- 구현의 편의상 입력 받는 정보의 길이는 아래의 “**각 메뉴 기능 수행과 관련한 제한사항**”에 명시된 최대치보다 항상 작은 값을 입력 받는 것으로 생각한다. 즉, 회원 아이디 길이, 등록하는 회원의 수 등은 매 입력 시 마다 검사할 필요가 없다.

각 메뉴 기능 수행과 관련한 제한 사항
-----------------------

- |  |
|--|
| 1) 저장 가능한 최대 회원 수: 50<br>2) 저장 가능한 최대 투표 수: 100<br>3) 각 투표별 최대 투표 항목 수: 10<br>4) RRN, ID, Password, 투표 주제 최대 길이: 32<br>5) 커맨드 로그 파일 이름: "commandLog.txt" |
|--|

각각의 메뉴는 다음 과 같은 기능을 수행한다.

각 메뉴별 수행 기능
-------------

<b>1. Register as a Member</b>
--------------------------------

- |  |
|--|
| 1) 입력 정보:<br>- 주민등록 번호, ID, 비밀번호<br>2) Valid case:<br>- 현재 로그인 되어 있지 않은 사용자에게 의해 요청되는 경우<br>- 입력된 RRN, ID 값 모두 중복되지 않는 경우(둘 중 하나라도 중복 될 경우, 실패)<br>3) Failure case:<br>- 현재 로그인 된 사용자에게 의해 요청되는 경우<br>- 입력된 RRN, ID 값 중 하나라도 중복 될 경우 |
|--|

## **2. Unsubscribe from System**

### 1) 입력 정보:

- ID, 비밀번호

### 2) Valid case:

- 현재 로그인 된 사용자에게 의해 요청 되는 경우
- 입력된 ID, 비밀번호가 현재 로그인 된 사용자의 ID, 비밀번호와 일치하는 경우

### 3) Failure case

- 현재 로그인 되어있지 않은 사용자에게 의해 요청 되는 경우.
- 현재 가입 되어있지 않은 입력 정보 일 경우.

## **3. Login**

### 1) 입력 정보:

- ID, 비밀번호

### 2) Valid case:

- 로그인 되어있지 않은 회원에 의해 입력된 ID, 비밀번호가 저장된 정보와 일치하는 경우

### 3) Failure case:

- 만약 비밀번호가 일치하지 않거나 아이디가 존재 하지 않는 회원일 경우.

## **4. Logout**

### 1) 입력 정보:

- 없음

### 2) Valid case:

- 로그인 된 사용자에게 의해 요청 되는 경우.

### 3) Failure case:

- 만약 로그인 되어있지 않은 사용자에게 의해 요청 되는 경우.

## **5. Delete Existing Vote**

### 1) 입력 정보:

- 투표 주제

### 2) Valid Case:

- 삭제를 요청한 회원이 로그인 되어 있는 경우
- 해당 투표가 존재하는 경우

### 3) Failure Case:

- 삭제 요청한 투표 주제가 존재하지 않는 경우
- 삭제 요청한 사용자가 로그인 되어있지 않는 경우

## **6. Add Vote**

### 1) 입력 정보:

- 투표 주제, 투표 항목 수

### 2) Valid case:

- 현재 로그인 되어있는 회원에 의해 요청 되는 경우.

### 3) Failure case:

- 동일한 투표 주제가 존재할 경우

## **7. View All Existing Votes**

### 1) 입력 정보:

- 없음

### 2) Valid case:

- 현재 로그인 멤버에 의해 요청 되는 경우

### 3) Failure case:

- 현재 로그인 되어있지 않은 멤버에 의해 요청 되는 경우

### 4) 출력 정보:

현재 시스템 상에 존재하는 모든 투표에 대한 투표 주제, 투표 현황을 화면상에 출력한다.

## **8. Cast a Vote**

### 1) 입력 정보:

- 투표 주제
- 투표할 항목 번호(항목 번호에 대한 득표 수가 증가된다.)

### 2) Valid case:

- 현재 로그인 된 사용자에게 의해 요청되는 경우
- 투표 주제가 존재하는 투표인 경우

### 3) Failure case:

- 현재 로그인 되어있지 않은 사용자에게 의해 요청되는 경우
- 입력 받은 투표 주제가 존재하지 않는 경우
- 이미 투표한 회원이 재투표 하는 경우
- 탈퇴한 회원이 동일한 주민번호로 다시 재가입 하여 재투표 하는 경우.

동일한 투표에 대해 한 사람이 계속해서 투표하는 것을 방지하기 위해, 시스템은 투표한 회원들의 주민번호를 저장하며, 이는 해당 회원이 탈퇴해도 지워지지 않고, 투표 기능 수행시 요청한 회원의 주민번호를 확인하여 중복 투표 여부를 검사한다.

## **9. Program Exit**

### 1) 입력 정보:

- 없음
- 2) Valid case:
- 모든 케이스에 가능

### 2.3 메뉴 동작 방식

1) 시스템은 텍스트 기반 UI에 의해 작동되며, 유저는 해당 메뉴 번호를 입력하여 메뉴를 선택할 수 있다.

```
=====
Num: 1
=====
1. Register as a Member
2. Unsubscribe from System
3. Login
4. Logout
5. Delete Existing Vote Item
6. Add a New Vote Item
7. List All Vote Items
8. Cast a Vote
9. Program Exit
=====
Select Menu:1
Selected Menu: 1. Register as a Member
1. Resident Registration Number: 1111
ID: id1
Password:pwd1
Registration Done!

=====
Num: 2
=====
1. Register as a Member
2. Unsubscribe from System
3. Login
4. Logout
5. Delete Existing Vote Item
6. Add a New Vote Item
7. List All Vote Items
8. Cast a Vote
9. Program Exit
=====
Select Menu:
```

2) 해당 메뉴가 선택되면(예시: 1. Register as Member), 해당 메뉴가 선택되었다는 메시지와 함께, 각 메뉴 별 필요한 정보(4.2 로그파일 형식 참조)를 입력 받도록 한다. 메뉴 디자인은 임의로 변경이 가능하나, 반드시 메뉴 번호, 순서, 번호별 기능은 기존과 같게 하도록 한다.

3) 해당 입력 정보가 유효한 정보일 경우, 성공메시지를 출력하고 다시 메뉴 선택으로 돌아간다. 만약 유효하지 않은 상태이거나, 해당 정보가 유효하지 않은 경우, 화면 상에 실패 메시지를

출력한다. 예를 들어, 현재 로그인 된 사용자가, 다시 로그인을 수행하거나, 회원 가입 시 중복된 정보를 입력하는 경우, 실패메시지를 출력한다. (4.2 로그파일 형식 참고)

```
=====
Num: 1
=====
1. Register as a Member
2. Unsubscribe from System
3. Login
4. Logout
5. Delete Existing Vote Item
6. Add a New Vote Item
7. List All Vote Items
8. Cast a Vote
9. Program Exit
=====
Select Menu:1
Selected Menu: 1. Register as a Member
Resident Registration Number:1111
ID:id1
Password:pwd1
Registration Done!
```

```
=====
Num: 2
=====
1. Register as a Member
2. Unsubscribe from System
3. Login
4. Logout
5. Delete Existing Vote Item
6. Add a New Vote Item
7. List All Vote Items
8. Cast a Vote
9. Program Exit
=====
Select Menu:1
Selected Menu: 1. Register as a Member
Resident Registration Number:1111
ID:id1
Password:pwd1
Registration Fail
```

Figure 2 회원 가입 실패 화면

```
=====
Num: 1
=====
1. Register as a Member
2. Unsubscribe from System
3. Login
4. Logout
5. Delete Existing Vote Item
6. Add a New Vote Item
7. List All Vote Items
8. Cast a Vote
9. Program Exit
=====
Select Menu:1
Selected Menu: 1. Register as a Member
Resident Registration Number:1111
ID:id1
Password:pwd1
Registration Done!
```

```
=====
Num: 2
=====
1. Register as a Member
2. Unsubscribe from System
3. Login
4. Logout
5. Delete Existing Vote Item
6. Add a New Vote Item
7. List All Vote Items
8. Cast a Vote
9. Program Exit
=====
Select Menu:3
Selected Menu: 3. Login
ID:id1
Password:pwd1
Login Success!
```

Figure 3 로그인 성공 화면

4) 메뉴 선택 -> 해당 기능 수행을 반복 수행하며, 9. Program Exit 메뉴 선택 시 프로그램을 종료한다.

## 5) 전체 실행 화면 예시

```
$ ./example.exe
Load log file?(1:yes, 0:no (log will be deleted)):
1

=====
Num: 1
=====
1. Register as a Member
2. Unsubscribe from System
3. Login
4. Logout
5. Delete Existing Vote Item
6. Add a New Vote Item
7. List All Vote Items
8. Cast a Vote
9. Program Exit
=====
Select Menu:1
Selected Menu: 1. Register as a Member
1. Resident Registration Number: 1111
ID: id1
Password:pwd1
Registration Done!

=====
Num: 2
=====
1. Register as a Member
2. Unsubscribe from System
3. Login
4. Logout
5. Delete Existing Vote Item
6. Add a New Vote Item
7. List All Vote Items
8. Cast a Vote
9. Program Exit
=====
Select Menu:1
Selected Menu: 1. Register as a Member
1. Resident Registration Number: 1112
ID: id2
Password:pwd2
Registration Done!

=====
Num: 3
=====
1. Register as a Member
2. Unsubscribe from System
3. Login
4. Logout
5. Delete Existing Vote Item
6. Add a New Vote Item
```



```
7. List All Vote Items
8. Cast a Vote
9. Program Exit
=====
Select Menu:3
Selected Menu: 3. Login
ID: id1
Password: pwd1
Login Success!

=====
Num: 4
=====
1. Register as a Member
2. Unsubscribe from System
3. Login
4. Logout
5. Delete Existing Vote Item
6. Add a New Vote Item
7. List All Vote Items
8. Cast a Vote
9. Program Exit
=====
Select Menu:6
Selected Menu: 6. Add a New Vote Item
Vote Subject: vote1
Number of Vote Items: 4
Vote Added!

=====
Num: 5
=====
1. Register as a Member
2. Unsubscribe from System
3. Login
4. Logout
5. Delete Existing Vote Item
6. Add a New Vote Item
7. List All Vote Items
8. Cast a Vote
9. Program Exit
=====
Select Menu:8
Selected Menu: 8. Cast a Vote
Vote Subject: vote1
Vote for: 1
Request processed successfully!

=====
Num: 6
=====
1. Register as a Member
2. Unsubscribe from System
3. Login
4. Logout
```

5. Delete Existing Vote Item
6. Add a New Vote Item
7. List All Vote Items
8. Cast a Vote
9. Program Exit

=====

Select Menu:8

Selected Menu: 8. Cast a Vote

Vote Subject: vote1

Vote for: 2

Vote casting failed!

=====

Num: 7

=====

1. Register as a Member
2. Unsubscribe from System
3. Login
4. Logout
5. Delete Existing Vote Item
6. Add a New Vote Item
7. List All Vote Items
8. Cast a Vote
9. Program Exit

=====

Select Menu:7

Selected Menu: 7. List All Vote Items.

Vote Subject: vote1, Vote Counts per Item - item1: 1 item2: 0 item3: 0 item4: 0

=====

Num: 8

=====

1. Register as a Member
2. Unsubscribe from System
3. Login
4. Logout
5. Delete Existing Vote Item
6. Add a New Vote Item
7. List All Vote Items
8. Cast a Vote
9. Program Exit

=====

Select Menu:5

Selected Menu: 5. Delete Existing Vote Item

Vote Subject to Delete: vote1

Vote Deleted!

=====

Num: 9

=====

1. Register as a Member
2. Unsubscribe from System
3. Login
4. Logout
5. Delete Existing Vote Item

```
6. Add a New Vote Item
7. List All Vote Items
8. Cast a Vote
9. Program Exit
=====
Select Menu:7
Selected Menu: 7. List All Vote Items.

=====
Num: 10
=====
1. Register as a Member
2. Unsubscribe from System
3. Login
4. Logout
5. Delete Existing Vote Item
6. Add a New Vote Item
7. List All Vote Items
8. Cast a Vote
9. Program Exit
=====
Select Menu:5
Selected Menu: 5. Delete Existing Vote Item
Vote Subject to Delete: vote2
Request Failed

=====
Num: 11
=====
1. Register as a Member
2. Unsubscribe from System
3. Login
4. Logout
5. Delete Existing Vote Item
6. Add a New Vote Item
7. List All Vote Items
8. Cast a Vote
9. Program Exit
=====
Select Menu:4
Selected Menu: 4. Logout
Logout Success

=====
Num: 12
=====
1. Register as a Member
2. Unsubscribe from System
3. Login
4. Logout
5. Delete Existing Vote Item
6. Add a New Vote Item
7. List All Vote Items
8. Cast a Vote
```

9. Program Exit

=====

Select Menu:3

Selected Menu: 3. Login

ID: id2

Password: pwd2

Login Success!

=====

Num: 13

=====

1. Register as a Member
2. Unsubscribe from System
3. Login
4. Logout
5. Delete Existing Vote Item
6. Add a New Vote Item
7. List All Vote Items
8. Cast a Vote
9. Program Exit

=====

Select Menu:8

Selected Menu: 8. Cast a Vote

Vote Subject: vote1

Vote for: 2

Vote casting failed!

=====

Num: 14

=====

1. Register as a Member
2. Unsubscribe from System
3. Login
4. Logout
5. Delete Existing Vote Item
6. Add a New Vote Item
7. List All Vote Items
8. Cast a Vote
9. Program Exit

=====

Select Menu:7

Selected Menu: 7. List All Vote Items.

=====

Num: 15

=====

1. Register as a Member
2. Unsubscribe from System
3. Login
4. Logout
5. Delete Existing Vote Item
6. Add a New Vote Item
7. List All Vote Items
8. Cast a Vote

```

9. Program Exit
=====
Select Menu:8
Selected Menu: 8. Cast a Vote
Vote Subject: vote1
Vote for: 2
Vote casting failed!

=====
Num: 16
=====
1. Register as a Member
2. Unsubscribe from System
3. Login
4. Logout
5. Delete Existing Vote Item
6. Add a New Vote Item
7. List All Vote Items
8. Cast a Vote
9. Program Exit
=====
Select Menu:

```

### 3. Class Design

#### 1) 클래스 모듈화

본 과제의 주요 목표인 클래스 설계를 달성하기 위해, 클래스 설계는 가능한 분리된 기능을 수행하도록 설계한다. 예를 들어, 하나의 클래스에 모든 기능(회원 정보를 담는 기능, 회원 리스트 관리 등) 수행하도록 하는 경우는 지양하도록 하도록 하며 이는 채점에 반영될 예정임.

#### 2) Class Design 예시

##### - Member Class

회원에 대한 정보를 담는 클래스

##### - Member Collection Class:

회원 리스트를 관리하는 클래스

-> 리스트 구현시 vector등 본인이 아는 STL 사용 가능, 채점시 테스트할 전체 회원의 수, 투표 수는 한정 되어있으므로, 배열을 사용하여 구현해도 무방함.

Member	MemberCollection
<b>멤버 변수</b> RRN;// 회원 주민번호 ...	<b>멤버 변수</b> // 멤버 리스트 배열 memberList[MAX_NUM_MEMBER]; ...
<b>멤버 함수</b> getLoginID( ); ... setPassword();	<b>멤버 함수</b> getMember ...

#### 4. Log 입출력

본 과제에서는 현재 시스템 상에 기록된 정보를 기록하기 위해 텍스트 형태의 로그 파일이 이용된다. 로그 파일은 입력되었던 커맨드들을 기록 및 읽어 들이는 "commandLog.txt" 파일이 있으며, 매번 메뉴를 선택하여 기능을 수행 할 때마다 로그를 남기도록 하여, 프로그램 종료 후 다시 실행 시켰을 때, 해당 지점에서 다시 시작 할 수 있도록 한다.

프로그램 실행 시, "Load log file?(1:yes, 0:no(log will be deleted))"라는 메시지와 함께, 커맨드 로그 파일 불러오기 여부를 결정할 수 있도록 한다. 만약 "0"을 입력할 시, 저장되었던 커맨드 로그파일은 지워지고 standard input으로부터만 입력을 받는다. 만약 커맨드 로그 파일을 불러올 경우, 로그 파일에 기록된 커맨드 기록 내용이 자동으로 수행되며, 모든 로그를 읽어왔을 경우, 시스템은 다시 standard input으로부터 입력을 받도록 한다. 즉, 로그 파일의 모든 입력을 읽어온 뒤부터는 다시 키보드 입력만을 받도록 한다.

##### 1) 커맨드 로그 기록

commandLog.txt 예시
1 1111 id1 pwd1
1 1112 id2 pwd2

커맨드 로그 파일은, "[메뉴 번호] [입력 정보]" 순으로 입력 받으며 각각의 입력 정보는 띄어쓰기로 구분된다. 예를 들어 "commandLog.txt"파일의 첫 번째 줄의 "1 1111 id1 pwd1"은 메뉴1 선택, 주민등록번호, ID, 비밀번호 입력을 의미한다. 별도의 테스트 입력 커맨드 로그 파일을 사용하여 채점에 활용할 계획이므로, **반드시 입력 정보는, 4.2 로그파일 형식을 참고하여 같은 순서로 입력 받도록 한다. 만약 이를 지키지 않아 프로그램작동 에러가 나는 경우 프로그램**

**실행 점수 0점으로 간주 된다.** 예를 들어, 1. 회원 가입 메뉴 기능 수행은, Resident Registration Number(RRN), ID, Password순으로 입력 받도록 되어 있고, 입력 로그에도 동일한 순서로 기록 되어야한다. **마지막 “9. Program Exit” 에 대한 로그는 기록하지 않는다.**

#### 4.1 로그 입출력 예시

다음은 아래와 같은 입력 커맨드 로그파일을 읽어 들인 후, 로그에 기록된 메뉴 기능을 자동으로 수행한 화면의 예이다.

commandLog.txt
1 1111 id1 pwd1
1 1112 id2 pwd2

```
$ ./example.exe
Load log file?(1:yes, 0:no (log will be deleted)):
1
```

```
=====
```

```
Num: 1
```

```
=====
```

```
1. Register as a Member
2. Unsubscribe from System
3. Login
4. Logout
5. Delete Existing Vote Item
6. Add a New Vote Item
7. List All Vote Items
8. Cast a Vote
9. Program Exit
```

```
=====
```

```
Select Menu:1
```

```
Selected Menu: 1. Register as a Member
```

```
1. Resident Registration Number: 1111
```

```
ID: id1
```

```
Password:pwd1
```

```
Registration Done!
```

```
=====
```

```
Num: 2
```

```
=====
```

```
1. Register as a Member
2. Unsubscribe from System
3. Login
4. Logout
5. Delete Existing Vote Item
6. Add a New Vote Item
7. List All Vote Items
8. Cast a Vote
9. Program Exit
```

```
=====
```

```
Select Menu:1
```

```
Selected Menu: 1. Register as a Member  
1. Resident Registration Number: 1112  
ID: id2  
Password:pwd2  
Registration Done!
```

```
=====  
Num: 3  
=====  
1. Register as a Member  
2. Unsubscribe from System  
3. Login  
4. Logout  
5. Delete Existing Vote Item  
6. Add a New Vote Item  
7. List All Vote Items  
8. Cast a Vote  
9. Program Exit  
=====  
Select Menu:9
```

Figure 3 실행 결과



## 4.2 로그 파일 형식

커맨드 로그파일은 다음과 같은 형식으로 기록되고 시스템 상에서도 다음에 명시된 순서대로 입력을 받는다. 커맨드 로그파일에 있는 각 정보 입력 순서를 반드시 지켜야한다.

"commandLog.txt" 형식
1 [주민번호] [ID] [비밀번호]
2 [ID] [비밀번호]
3 [ID] [비밀번호]
4
5 [투표 주제]
6 [투표 주제] [항목 수]
7
8 [투표 주제] [항목 번호]

## 4.3 로그 입출력 함수 사용법

다음은 기본으로 제공되는 소스 파일의 일부이다. 파일 입출력은 ofstream과 ifstream을 사용하여 구현한다.

```
ifstream inLogRead; // commandLog.txt 파일로부터 커맨드 정보를 읽는 객체
ofstream inLogWrite; // commandLog.txt에 현재까지 수행된 커맨드를 기록하는 객체
```

### 1) 로그 파일 열기

커맨드를 stdin으로부터 받는 경우
inLogWrite.open(COMMAND_LOG_FILE_NAME, ios::out); //현재 까지 입력된 커맨드를 기록하는 객체
커맨드를 파일로부터 읽는 경우
inLogRead.open(COMMAND_LOG_FILE_NAME, ios::in); //read mode inLogWrite.open(COMMAND_LOG_FILE_NAME, ios::in ios::app); //concat without truncation

각각의 객체에는 open()이라는 멤버 함수가 존재하고, open할 파일명, open할 mode를 함수 인자로 전달할 수 있다. ios::out 과 ios::in 모드는 각각 출력 mode, 입력 mode를 의미하며,

ios::in|ios::app 모드는 현재 기록된 파일 내용을 삭제하지 않고, 항상 파일 끝에 append하기 위한 mode를 의미한다.

## 2) 로그 read

다음은 커맨드 로그를 파싱하기 위한 코드의 일부이다. 각각의 메뉴 선택 시 다음과 같은 메뉴 파싱 절차가 진행된다.

1. getline() 함수를 통해 inLogRead에 의해 open된 commandLog.txt파일로부터, 한 줄을 읽어온다.
2. stringstream객체인 stream의 str()멤버 함수를 통해 현재 stream의 내용을 해당 line으로 설정한다.
3. while문을 돌면서, 공백문자, " "에 의해 구분된 token들을 tokens배열에 저장한다.
4. stream의 내용을 초기화 한다.
5. tokens 배열에 메뉴 기능에 필요한 정보들이 저장되어 있고 이를 메뉴 기능 수행에 활용한다.

```
string tokens[4], line; stringstream stream; int nTokens;
...
getline(inLogRead, line);
stream.str(line);
nTokens = 0;
clearTokens(tokens); //initializing array of strings
//Read all tokens(seperated by " ")
while (stream >> tokens[nTokens]) nTokens++;
stream.clear();
menu = atoi(tokens[0].c_str());
...
```

## 3) 로그 write

```
string RRN, ID, password;
...
inLogWrite << "1 " << RRN << " " << ID << " " << password << " " << endl;
...
```

ofstream 객체와 "<<" 연산자를 활용하여 출력하고자 하는 내용을 파일로 출력한다.

## 5. 기타

1) 구현하지 못한 기능이 있다면 로그 파일상에 있는 모든 커맨드 라인들을 읽어 들일 수는 있게 구현하며, 구현하지 못한 기능에 있어서는 다음의 예시와 같이 "Not Implemented" 를 화면상에 출력하도록 한다.

```
$ ./example.exe
Load log file?(1:yes, 0:no (log will be deleted)):1
```

```
=====
```

```
Num: 1
```

```
=====
```

1. Register as a Member
2. Unsubscribe from System
3. Login
4. Logout
5. Delete Existing Vote Item
6. Add a New Vote Item
7. List All Vote Items
8. Cast a Vote
9. Program Exit

```
=====
```

```
Select Menu:1
```

```
Selected Menu: 1. Register as a Member
```

```
1. Resident Registration Number: 1111
```

```
ID: id1
```

```
Password:pwd1
```

```
Not Implemented
```

```
=====
```

```
Num: 2
```

```
=====
```

1. Register as a Member
2. Unsubscribe from System
3. Login
4. Logout
5. Delete Existing Vote Item
6. Add a New Vote Item
7. List All Vote Items
8. Cast a Vote
9. Program Exit

```
=====
```

```
Select Menu:1
```

```
Selected Menu: 1. Register as a Member
```

```
1. Resident Registration Number: 1112
```

```
ID: id2
```

```
Password:pwd2
```

```
Not Implemented
```

```
=====
```

```
Num: 3
```

```

=====
1. Register as a Member
2. Unsubscribe from System
3. Login
4. Logout
5. Delete Existing Vote Item
6. Add a New Vote Item
7. List All Vote Items
8. Cast a Vote
9. Program Exit
=====
Select Menu:0
Undefined menu selection

=====
Num: 4
=====
1. Register as a Member
2. Unsubscribe from System
3. Login
4. Logout
5. Delete Existing Vote Item
6. Add a New Vote Item
7. List All Vote Items
8. Cast a Vote
9. Program Exit
=====
Select Menu:9

```

2) 예시로 주어진 코드를 사용하지 않고 구현하여도 되고, 자유롭게 변경 가능하다. 반드시 **4.2 로그파일 형식은 지킬 수 있도록 한다.**

## 6. 과제 제출 관련

- 1) 제출 파일은 Assn2.zip, Assn2\_Report.pdf 형태로 두가지 파일을 따로 업로드 할 것.
- 2) 실행 파일 파일을 제외한 나머지 makefile, 소스 및 헤더 파일만을 제출 하고, 해당 압축파일을 풀었을 때 make 커맨드만으로 컴파일이 가능하도록 제출할 것.