 # Problem Statement:

Write a C++ program that creates an output file, writes information to it, closes the file, open it again as an i/p file & read the information file from the file.

 # Objectives : To learn the concept of file handling

 # Outcomes: Implement reading & writing to the file using OOP concepts.

~~Ha~~

 # Hardware requirements:

Manufacturer & Model: Acer Swift-3 (intel i5 8th gen)

Installed Memory: 8GB RAM, 512GB SSD

Architecture : 64-bit

 # Software Requirements:

Operating system: windows-10 home single languge.

IDE: codeblock version: 20.03

compiler: g++ (version: 10.1.0)

 # Theory:

stream: It is a sequence of bytes. If acts as sources from which the input data can be obtained @ as a destination to which the output data can be sent.

i/p stream: used to hold i/p from data producers.

output stream: used to hold o/p for particular data consumer.

data type	description
ofstream	represent output file stream & used to create files & to write information to files.
ifstream	represent input file stream & used to read info. from files.

| fstream | represent file system generally. |

file opening modes:

Mode: Flag	Description
ios: app	append mode.
ios: ate	open file & move control to end of file
ios: in	open file for reading
ios: out	open file for writing
ios: trunk	if file already exists, it's contents will be truncated before opening the file.

pseudo code:

writing to file:

1. open file in write mode
2. get information from user
3. write that information to file.
4. close the file.

Reading from file:

1. open file in read mode
2. iterate through each line of file.
3. print each line of file.
4. close the file.

~~App~~

Append to the file:

1. open file in append mode
2. get information from user
3. append information to file.
4. close the file.

The classes used in the program are from standard library. They are all file handling classes provided by C++.

 # Testcases
① Writing to the file.
② Reading from file. } the screenshot of output are attached.
③ Appending to the file

 # Conclusion
Hence, we have studied the concept of file handling.

Input and Output Demo:

Writing to the newly created file:

```
Enter
0 to exit
1 to write to a file
2 to append to file
3 to read from file
:1
How many users: 2

Enter name: Lanson
Enter email: lanson.gray@hotmail.com

Enter name: Marigrane
Enter email: mari.lasi@gmail.com
```
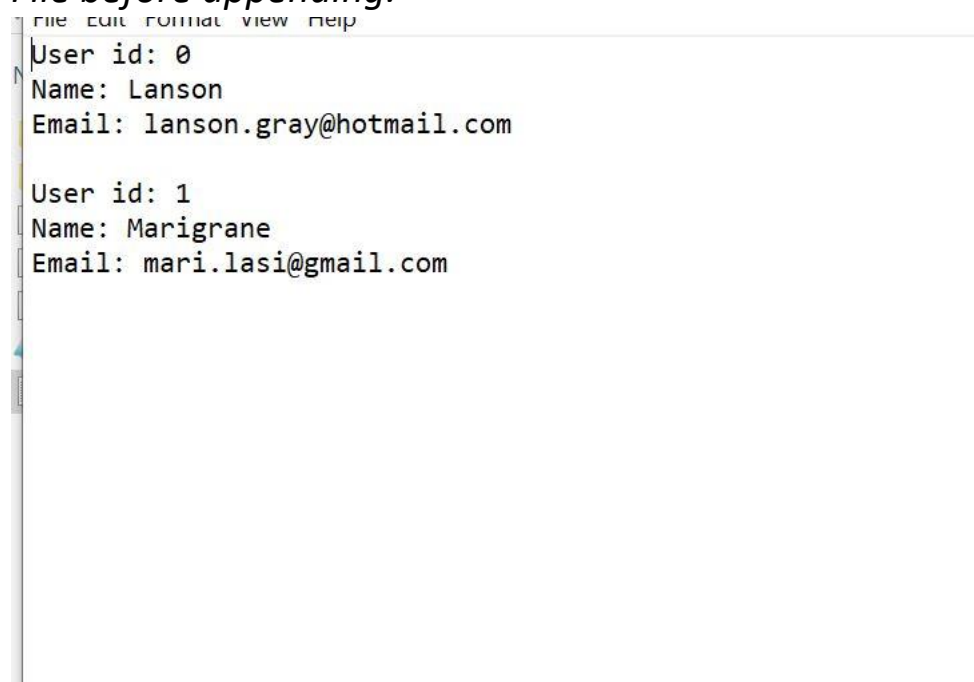
Appending to an existing file (appending to the same file created above):

```
Enter
0 to exit
1 to write to a file
2 to append to file
3 to read from file
:2
How many users: 1

Enter name: Tom
Enter email: tom.leinghton@mit.edu
```

File before appending:

```
File Edit Format View Help
User id: 0
Name: Lanson
Email: lanson.gray@hotmail.com

User id: 1
Name: Marigrane
Email: mari.lasi@gmail.com
```

File after appending:

```
User id: 0
Name: Lanson
Email: lanson.gray@hotmail.com

User id: 1
Name: Marigrane
Email: mari.lasi@gmail.com

User id: 2
Name: Tom
Email: tom.leinghton@mit.edu
```

Reading from file:

```
Enter
0 to exit
1 to write to a file
2 to append to file
3 to read from file
:3

~~~~~~~~~~~~Reading From the File~~~~~~~~~~~
User id: 0
Name: Lanson
Email: lanson.gray@hotmail.com

User id: 1
Name: Marigrane
Email: mari.lasi@gmail.com

User id: 2
Name: Tom
Email: tom.leinghton@mit.edu


Enter
```

Thank you.