Object Oriented Programming (Week 6)

2023

KWANGWOON UNIVERSITY

DEPT. OF COMPUTER ENGINEERING

Contents

- Assignment 2-2. 1
- Assignment 2-2. 2
- Assignment 2-2. 3
- Assignment 2-2. 4

ASSIGNMENT 2-2. 1

Assignment 2-2. 1

- (**Nested Array**) In script languages like Ruby or Python, arrays can contain values and other arrays. We will replicate this structure in C++. You are required to create an **array** that can contain two types of elements:

  1) **A string of up to 20 characters**

  2) **Another array**

  A string consists of one or more alphabets in uppercase or lowercase, and an array can be empty.

Assignment 2-2. 1

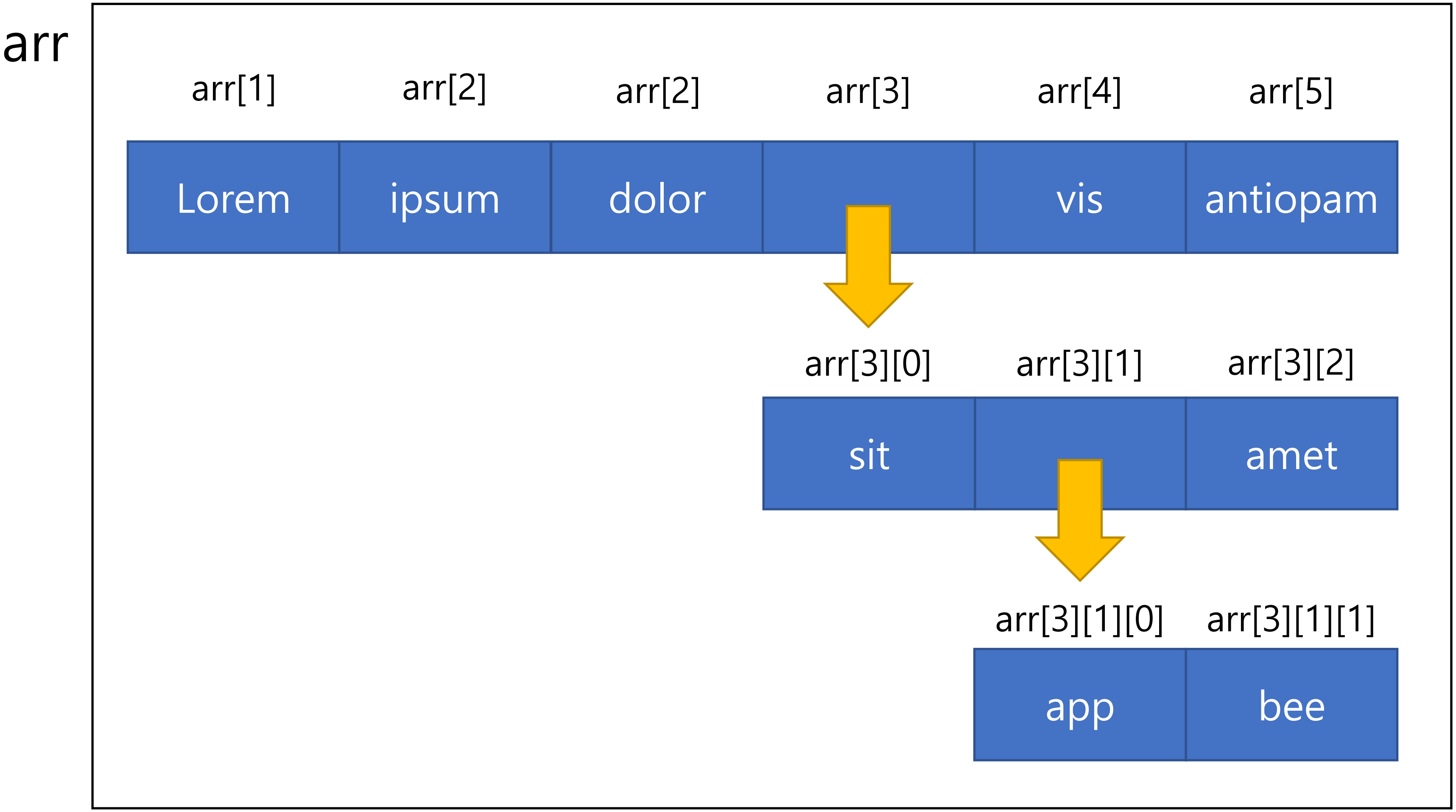
<Rules>

- The first line of the input contains **an array and its internal values**, and from the second line onwards, specify **which element to access**.
- If an input of the form "**arr[a][b][c]…**" is entered, **the string or array contained at that location** is output.
- If the element at that position is a **string**, print the string, and if it's an **array**, print the elements separated by commas and spaces.
- If the string "exit" is entered, exit the program.
- There may be zero or more spaces between the string, comma, and brackets in the input.
- Each input line contains at most 100,000 characters.

Input	Output
[Lorem, ipsum, dolor, [sit, amet], vis, antiopam]	[sit, amet]
arr[3]	[Lorem, ipsum, dolor, [sit, amet], vis, antiopam]
arr	amet
arr[3][1]	
exit	

Assignment 2-2. 1

[Lorem, ipsum, dolor, [sit, [app, bee], amet], vis, antiopam]



arr

arr[1] arr[2] arr[2] arr[3] arr[4] arr[5]

Lorem ipsum dolor vis antiopam

arr[3][0] arr[3][1] arr[3][2]

sit amet

arr[3][1][0] arr[3][1][1]

app bee

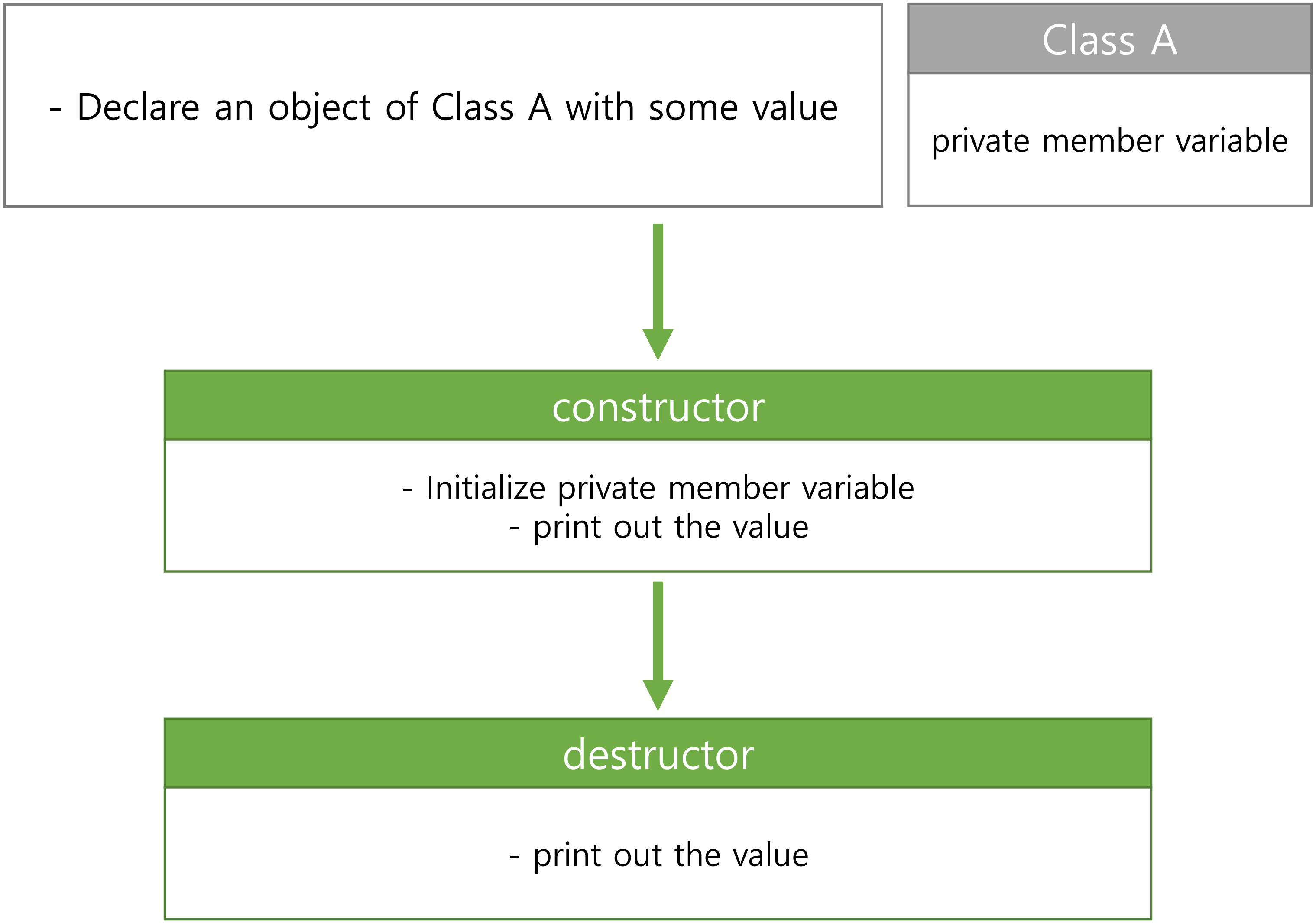
Assignment 2-2. 2

Assignment 2-2. 2

- (**A Class**) Implement a **Class A** with a **single private member variable of type double**. Class A should not have any other member functions except for the **constructor** and **destructor**. In the main function, declare an object of Class A, initialize its member variable by inputting a value, and print out the value when the object is created and destroyed, as shown in the example below. Please note that if you do not implement the class as required, no points will be given for the solution.

Input	Output
20	Class A is created with 20 Class A is destroyed with 20

Assignment 2-2. 2



- Declare an object of Class A with some value

Class A

private member variable

constructor

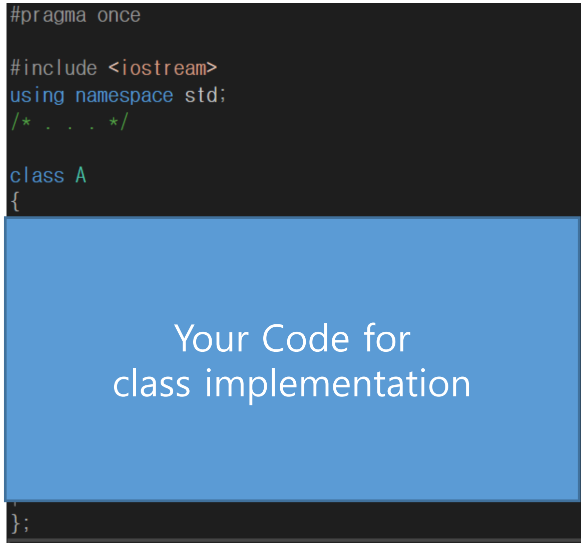
- Initialize private member variable
- print out the value

destructor

- print out the value

Assignment 2-2. 2

< A.h >

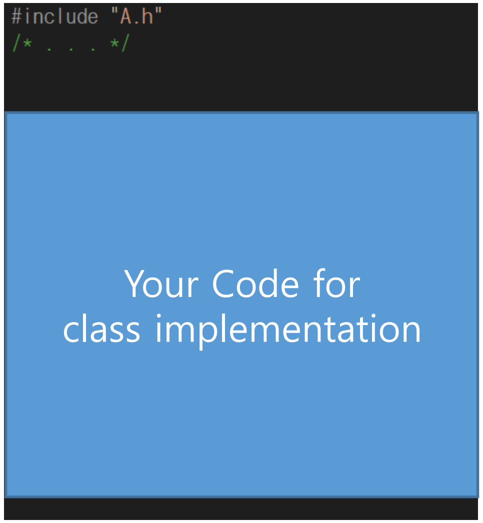


```
#pragma once

#include <iostream>
using namespace std;
/* . . . */

class A
{
```

Your Code for
class implementation

```
};
```

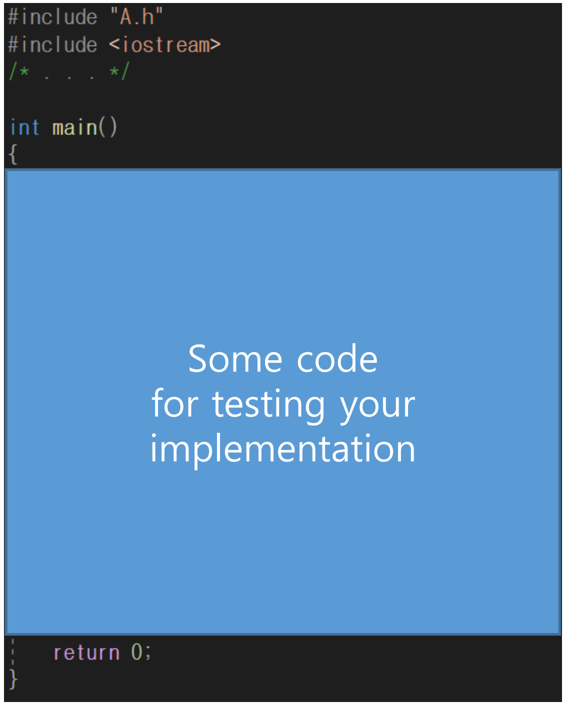
< A.cpp >



```
#include "A.h"
/* . . . */
```

Your Code for
class implementation

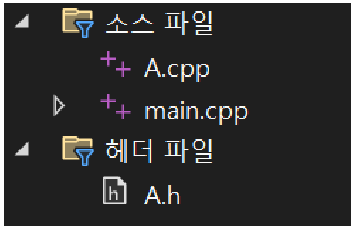
< main.cpp >



```
#include "A.h"
#include <iostream>
/* . . . */

int main()
{
```

Some code
for testing your
implementation

```
    return 0;
}
```



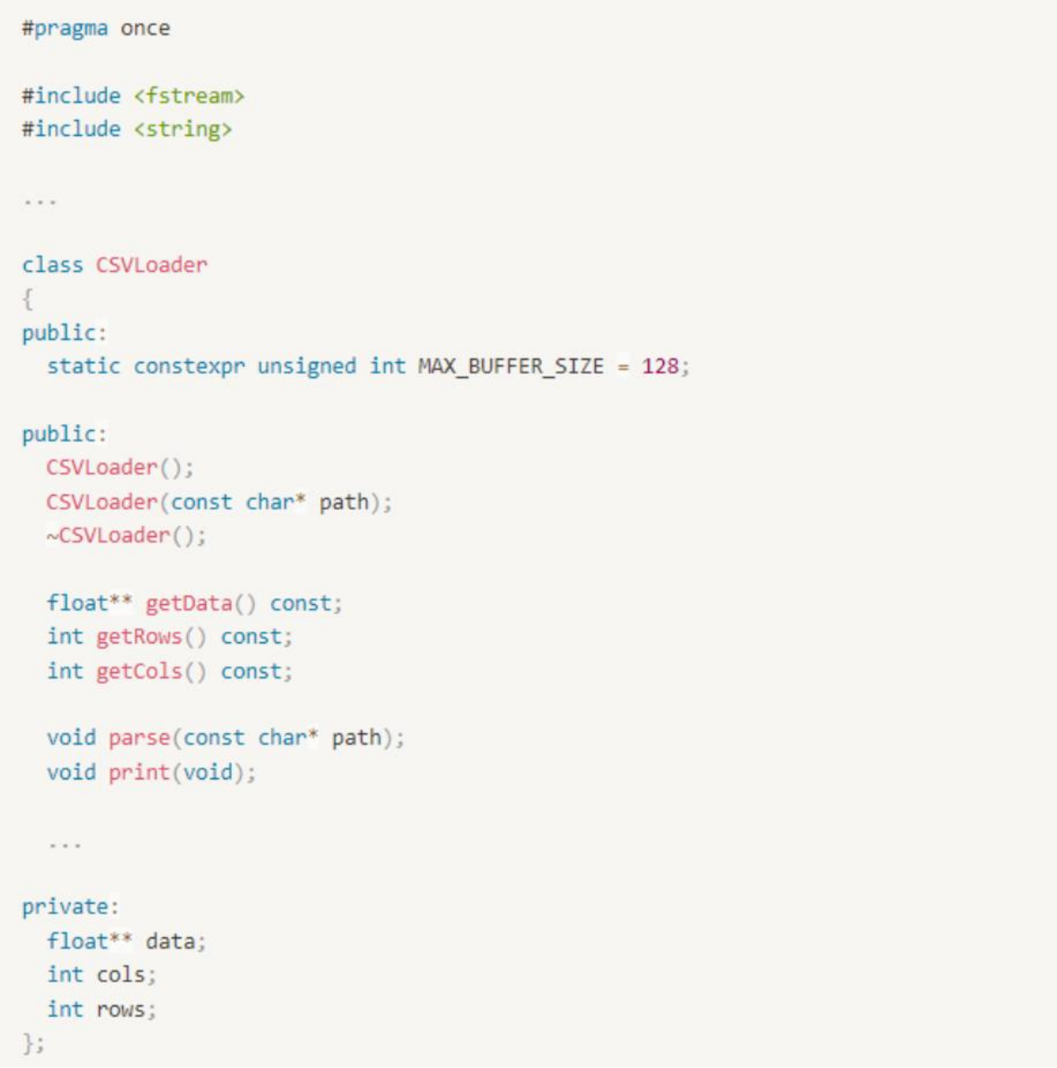
```
◢ 소스 파일
      A.cpp
  ▷   main.cpp
◢ 헤더 파일
      A.h
```

ASSIGNMENT 2-2. 3

Assignment 2-2. 3

- (**CSV Parsing**) Implement a class(**CSVLoader**) that parses **CSV**(Comma Separated Variables) **format** and stores the data in the desired type(**float**). Please note that the member variable **cols** and **rows** must be determined in Run-Time.

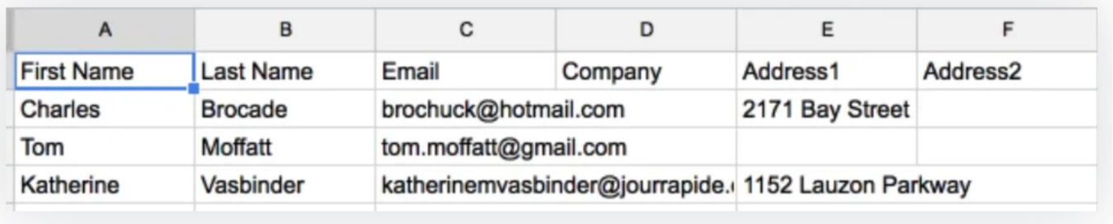
Assignment 2-2. 3



```
#pragma once

#include <fstream>
#include <string>

...

class CSVLoader
{
public:
  static constexpr unsigned int MAX_BUFFER_SIZE = 128;

public:
  CSVLoader();
  CSVLoader(const char* path);
  ~CSVLoader();

  float** getData() const;
  int getRows() const;
  int getCols() const;

  void parse(const char* path);
  void print(void);

  ...

private:
  float** data;
  int cols;
  int rows;
};
```

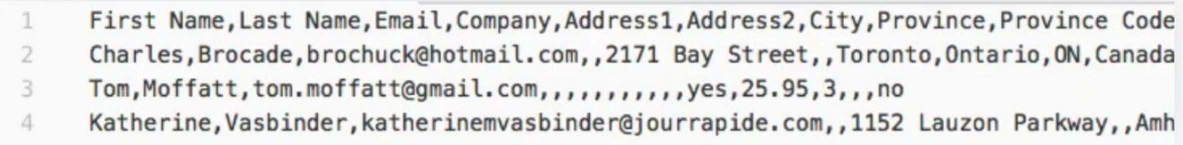
Assignment 2-2. 3

- CSV(Comma Separated Variables) format
  - 2차원 표 형태의 데이터를 저장하는 파일 형식
  - 구분자(,)로 열 구분 / 개행으로 행 구분
  - 모든 행은 같은 개수의 열을 가짐



A	B	C	D	E	F
First Name	Last Name	Email	Company	Address1	Address2
Charles	Brocade	brochuck@hotmail.com		2171 Bay Street	
Tom	Moffatt	tom.moffatt@gmail.com			
Katherine	Vasbinder	katherinemvasbinder@jourrapide.		1152 Lauzon Parkway	

< Excel format >

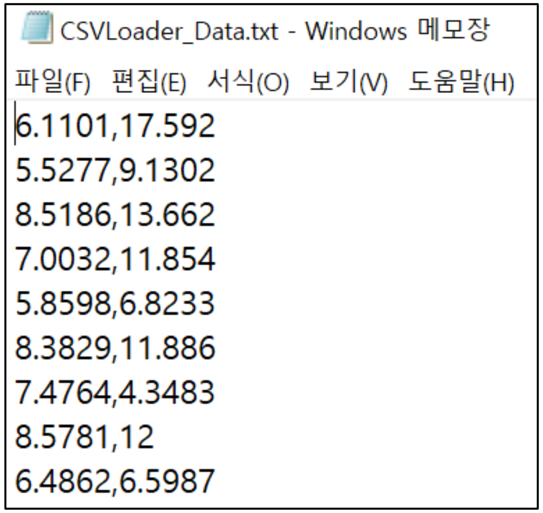


```
First Name,Last Name,Email,Company,Address1,Address2,City,Province,Province Code
Charles,Brocade,brochuck@hotmail.com,,2171 Bay Street,,Toronto,Ontario,ON,Canada
Tom,Moffatt,tom.moffatt@gmail.com,,,,,,,,,,yes,25.95,3,,,no
Katherine,Vasbinder,katherinemvasbinder@jourrapide.com,,1152 Lauzon Parkway,,Amh
```

< CSV format >

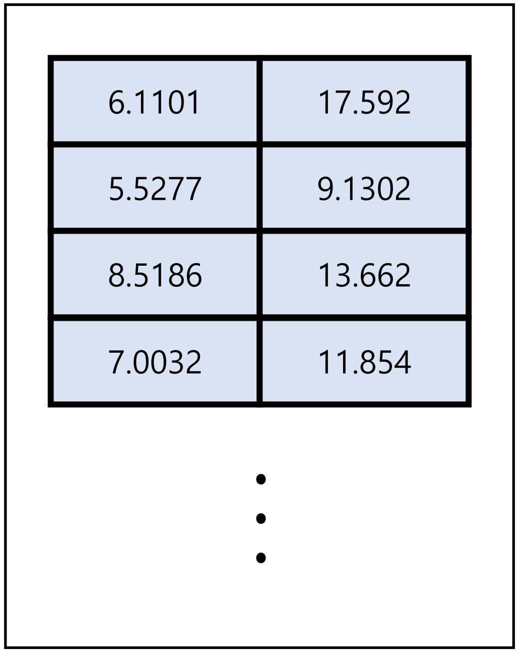
Assignment 2-2. 3

CSVLoader_Data.txt


CSVLoader_Data.txt - Windows 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

```
6.1101,17.592
5.5277,9.1302
8.5186,13.662
7.0032,11.854
5.8598,6.8233
8.3829,11.886
7.4764,4.3483
8.5781,12
6.4862,6.5987
```

CSVLoader's
member variable **data**
(cols = 2, row = 97)

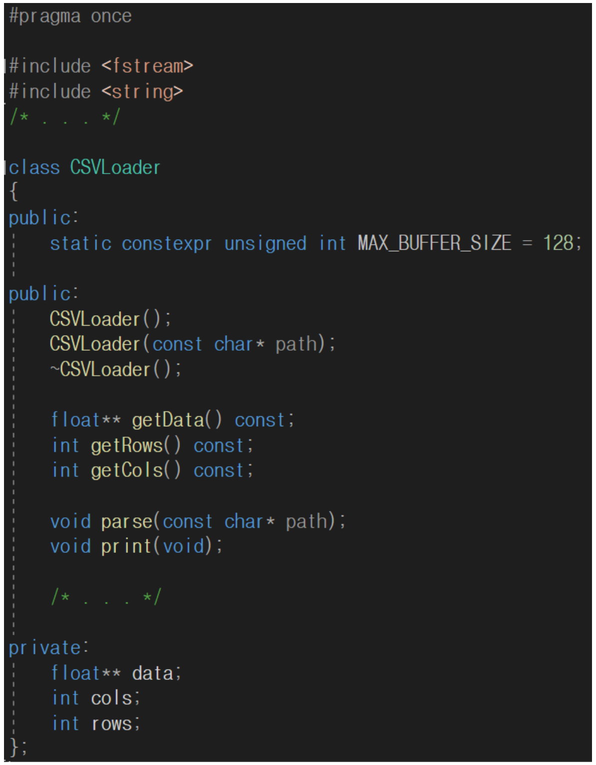


6.1101	17.592
5.5277	9.1302
8.5186	13.662
7.0032	11.854

•
•
•

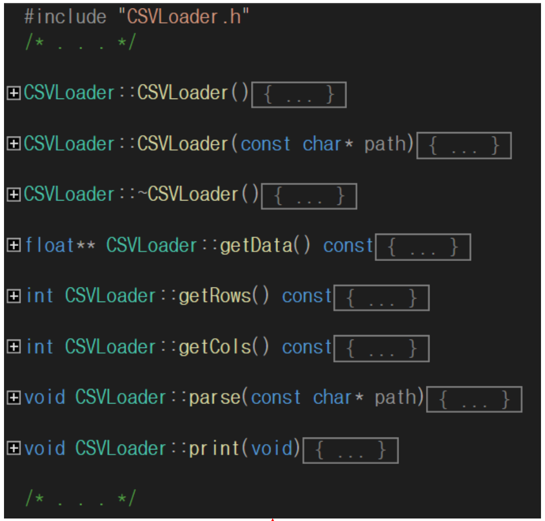
Assignment 2-2. 3

< CSVLoader.h >



```
#pragma once

#include <fstream>
#include <string>
/* . . . */

class CSVLoader
{
public:
    static constexpr unsigned int MAX_BUFFER_SIZE = 128;

public:
    CSVLoader();
    CSVLoader(const char* path);
    ~CSVLoader();

    float** getData() const;
    int getRows() const;
    int getCols() const;

    void parse(const char* path);
    void print(void);

    /* . . . */

private:
    float** data;
    int cols;
    int rows;
};
```

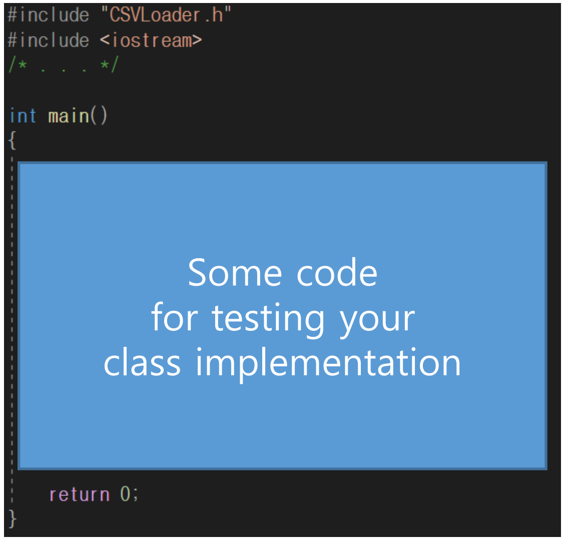
< CSVLoader.cpp >



```
#include "CSVLoader.h"
/* . . . */

⊞CSVLoader::CSVLoader(){ ... }

⊞CSVLoader::CSVLoader(const char* path){ ... }

⊞CSVLoader::~CSVLoader(){ ... }

⊞float** CSVLoader::getData() const{ ... }

⊞int CSVLoader::getRows() const{ ... }

⊞int CSVLoader::getCols() const{ ... }

⊞void CSVLoader::parse(const char* path){ ... }

⊞void CSVLoader::print(void){ ... }

/* . . . */
```

Class implementation code

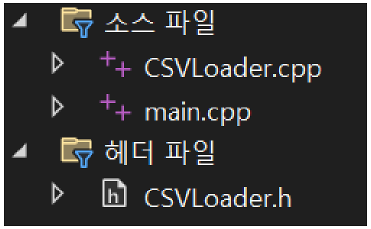
< main.cpp >



```
#include "CSVLoader.h"
#include <iostream>
/* . . . */

int main()
{
```

Some code
for testing your
class implementation

```
    return 0;
}
```



ASSIGNMENT 2-2. 4

Assignment 2-2. 4

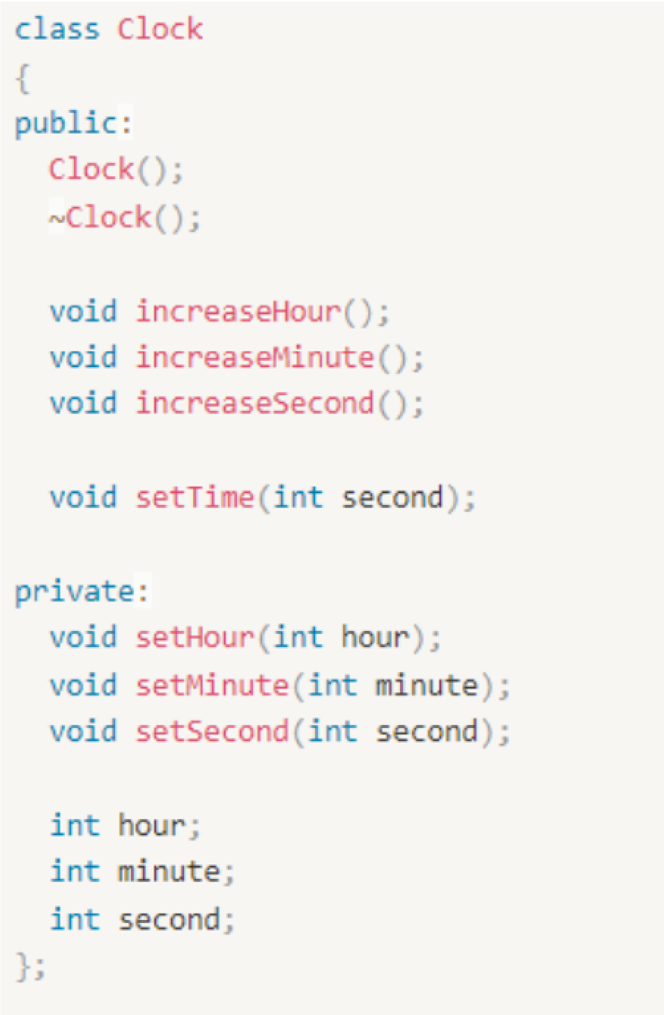
- (**Clock Class**) Implement a "**Clock**" class that contains three private integer member variables: **hour**, **minute**, and **second**. The "Clock" class has four public member functions: **setTime()**, **increaseSecond()**, **increaseMinute()**, **increaseHour().** **setTime()** takes seconds as input and then calls three private member functions sequentially to set second, minute, and hour.

  For Example, when **92200 seconds** is passed to setTime(), the second is set to 40 by setSecond(), and then minute is set to 36 by setMinute(), and an hour is set to 1 by setHour().

Assignment 2-2. 4

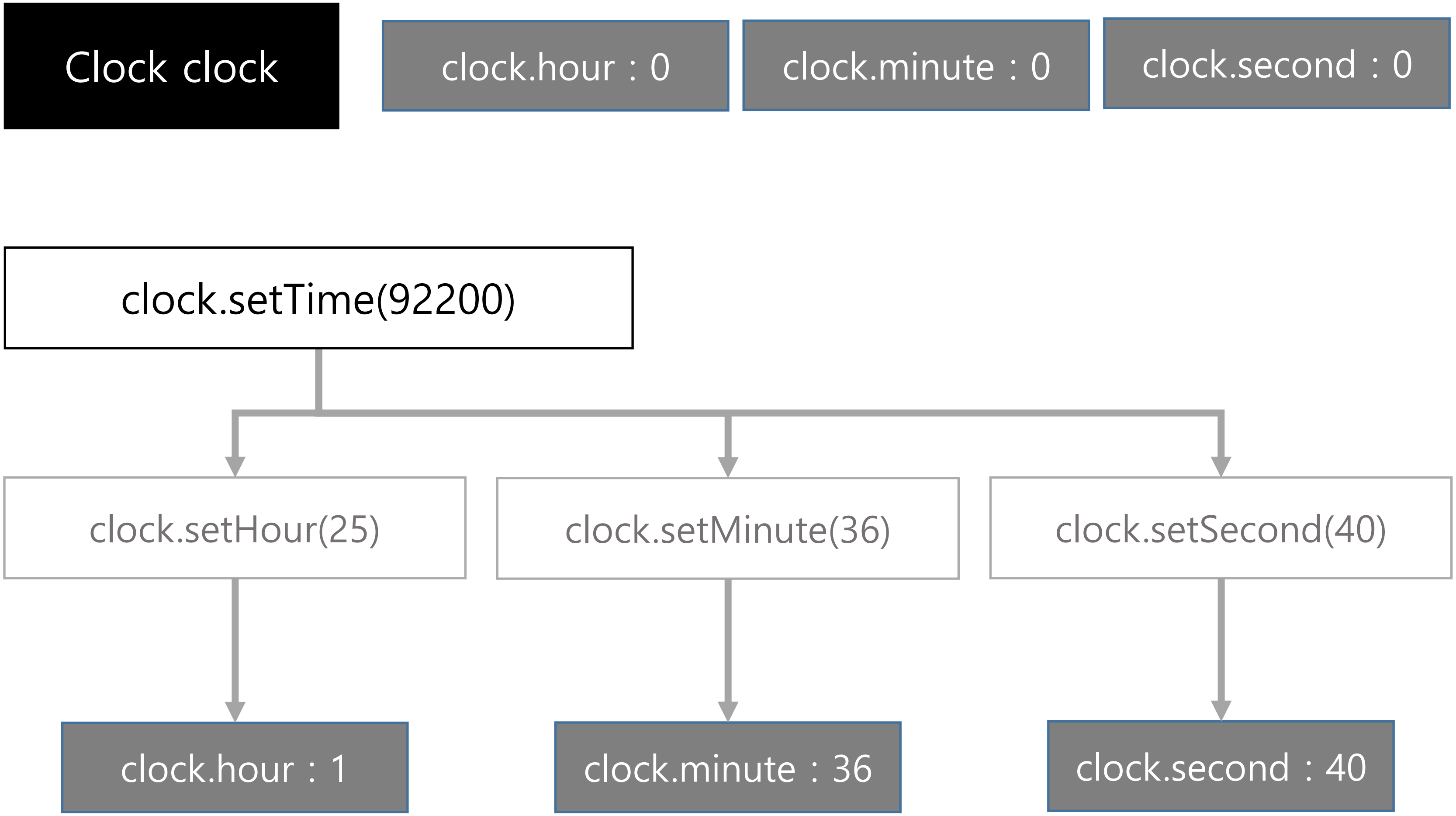
- The other public member functions: **increaseHour()**, **increaseMinute()**, and **increaseSecond()** increase their corresponding member variables.
  During increasing function is called if the second reaches 60 set the value to 0 and increase 1 minute and likewise if the minute reaches 60 set the value to 0 and increase 1 hour.
  When the hour reaches 24, then set it to 0.

Assignment 2-2. 4



```
class Clock
{
public:
  Clock();
  ~Clock();

  void increaseHour();
  void increaseMinute();
  void increaseSecond();

  void setTime(int second);

private:
  void setHour(int hour);
  void setMinute(int minute);
  void setSecond(int second);

  int hour;
  int minute;
  int second;
};
```

Assignment 2-2. 4



Clock clock

clock.hour : 0

clock.minute : 0

clock.second : 0

clock.setTime(92200)

clock.setHour(25)

clock.setMinute(36)

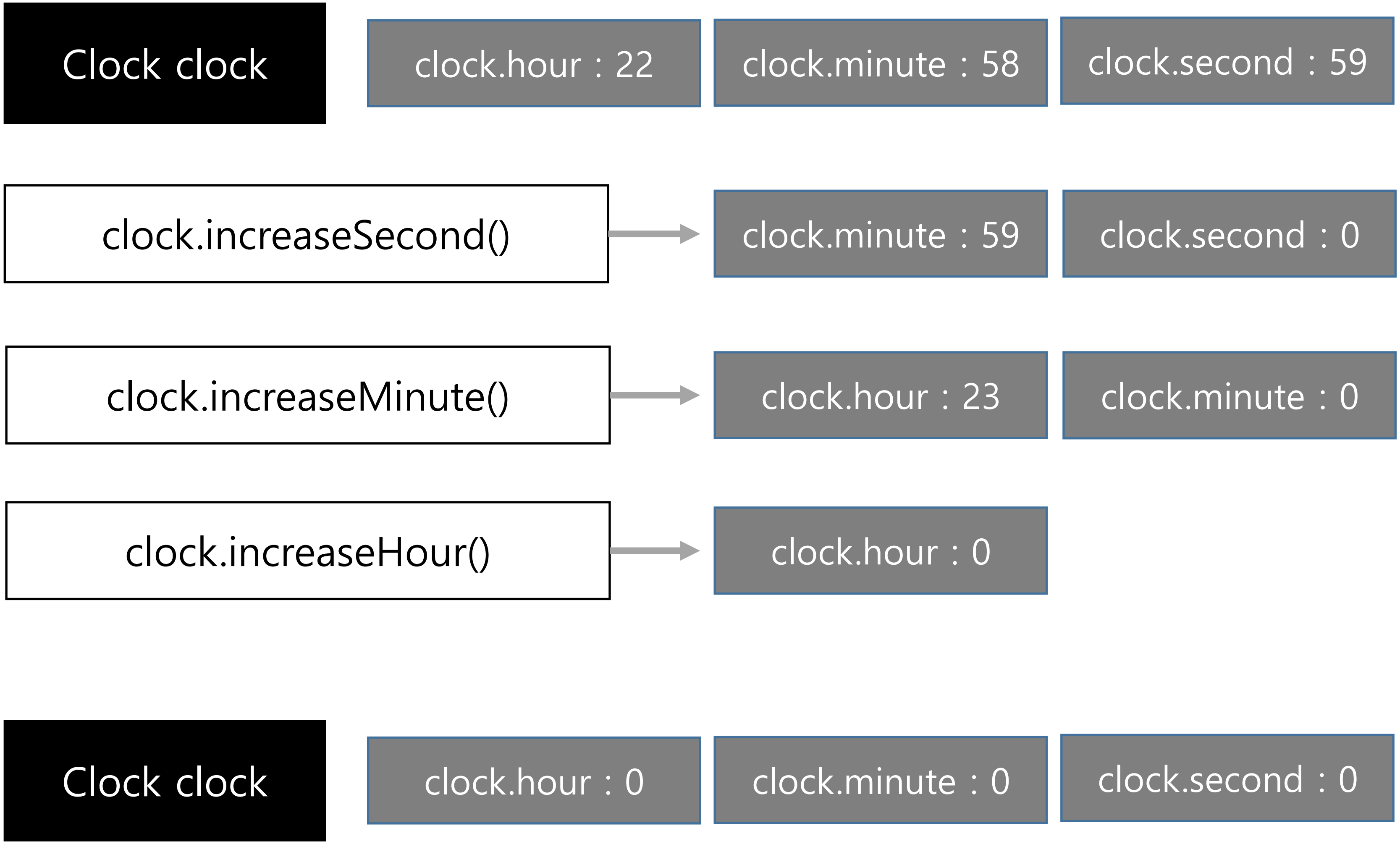
clock.setSecond(40)

clock.hour : 1

clock.minute : 36

clock.second : 40

Assignment 2-2. 4



Clock clock | clock.hour : 22 | clock.minute : 58 | clock.second : 59

clock.increaseSecond() → clock.minute : 59 | clock.second : 0

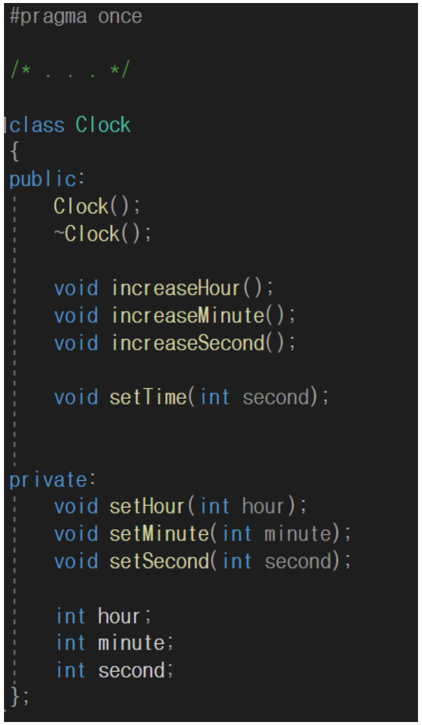
clock.increaseMinute() → clock.hour : 23 | clock.minute : 0

clock.increaseHour() → clock.hour : 0

Clock clock | clock.hour : 0 | clock.minute : 0 | clock.second : 0

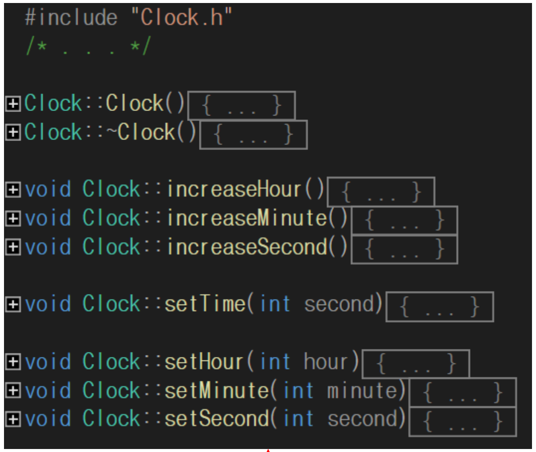
Assignment 2-2. 4

< Clock.h >



```
#pragma once

/* . . . */

class Clock
{
public:
    Clock();
    ~Clock();

    void increaseHour();
    void increaseMinute();
    void increaseSecond();

    void setTime(int second);


private:
    void setHour(int hour);
    void setMinute(int minute);
    void setSecond(int second);

    int hour;
    int minute;
    int second;
};
```

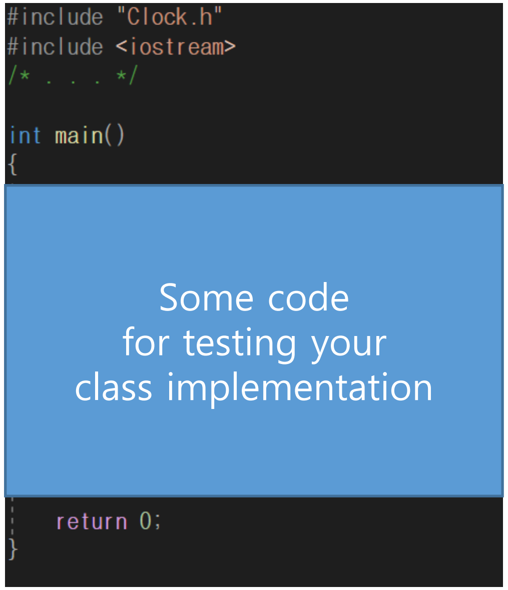
< Clock.cpp >



```
 #include "Clock.h"
 /* . . . */

⊞Clock::Clock() { ... }
⊞Clock::~Clock() { ... }

⊞void Clock::increaseHour() { ... }
⊞void Clock::increaseMinute() { ... }
⊞void Clock::increaseSecond() { ... }

⊞void Clock::setTime(int second) { ... }

⊞void Clock::setHour(int hour) { ... }
⊞void Clock::setMinute(int minute) { ... }
⊞void Clock::setSecond(int second) { ... }
```

Class implementation code

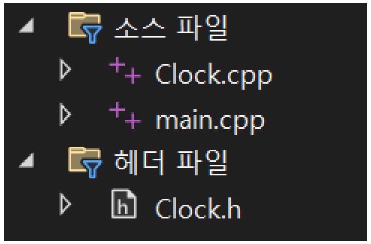
< main.cpp >



```
#include "Clock.h"
#include <iostream>
/* . . . */

int main()
{
```

Some code
for testing your
class implementation

```
    return 0;
}
```



과제 제출 방법

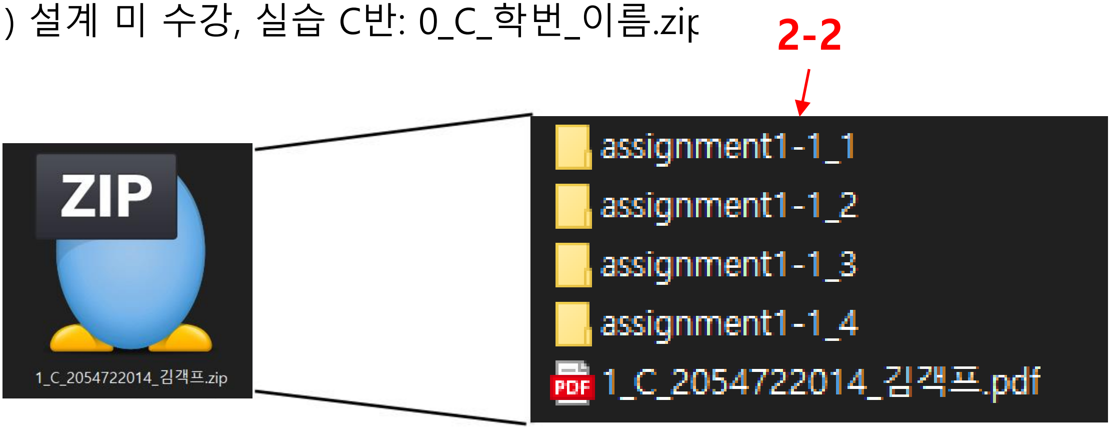
과제 제출 방법

- **FTP Upload (Klas 과제 제출 X)**
  - Address : ftp://223.194.8.1:1321
  - username : IPSL_OBJ
  - password : ipslobj_2023

- **Due date**
  - Soft copy: 마감일 4/14(금) 23:59:59까지 제출 (서버시간 기준)
  - Delay
    - 마감일 이후 +7일까지 제출 가능
    - 단, 1일 초과마다 과제 총점의 10%씩 감점

과제 제출 방법

- Soft copy
  - 과제(보고서, 소스 코드)를 압축한 파일 제출
    - 설계반_실습반_학번_이름.zip
      - 예) 설계1반 수강, 실습 A반: 1_A_학번_이름.zip
      - 예) 설계 수강, 실습 미수강: 2_0_학번_이름.zip
      - 예) 설계 미 수강, 실습 C반: 0_C_학번_이름.zip



1_C_2054722014_김객프.zip

assignment1-1_1
assignment1-1_2
assignment1-1_3
assignment1-1_4
1_C_2054722014_김객프.pdf

  - 과제 수정하여 업로드 시 버전 명시
    - 설계반_실습반_학번_이름_verX.zip

과제 제출 방법

- Soft copy
  - 과제 보고서
    - 영문 또는 한글로 작성
    - **반드시 PDF**로 제출 (PDF 외 파일 형식으로 제출시 0점 처리)
    - 보고서 양식
      - 문제 및 설명(문제 capture 금지) / 결과 화면 / 고찰
      - 보고서 양식은 아래 경로에서 참고
        - https://www.ipsl.kw.ac.kr/post/1%EC%B0%A8-%EA%B3%BC%EC%A0%9C
    - **소스코드 제외**
    - 분량 제한 없음
    - **표절 적발 시 0점 처리**
  - 소스 코드
    - Visual Studio 2022 community 사용 필수
      - https://docs.microsoft.com/ko-kr/visualstudio/install/install-visual-studio?view=vs-2022
    - STL (Standard Template Library) 사용 금지 (vector, map, algorithm 등)
    - Debug 폴더를 제외한 모든 파일 제출
      - .sln 파일 포함(.cpp 만 제출하지 말것)
    - **각 문제마다 프로젝트 파일 생성 필수**
    - **주석 반드시 달기**
    - **소스코드 표절 적발 시 0점 처리**

End of Presentation

Q&A