SMU | LYLE
SCHOOL OF ENGINEERING

CS 7345 ADVANCED APPLICATIONS: LAB 2

Qing Gao 48358925

Library lending sales statistics

This is a library lending sales statistics program, including library class and book class. Book class is used to store book information, and library class is used to store library management information. When a student borrows a book, the system will first judge whether there is this book and whether there is enough inventory of this book. If so, students need to pay the corresponding rent to terminate the contract, The program will count the total and average borrowing sales of the library.

1. Document API for library

a. Provide developer documentation for the API explaining each endpoints use and functionality. Be sure to explain fully the input/output for method, expected outcomes as well as any assumptions that are made about its use and/or data requirements.

book.h

```
class Book
{
public:
    Book();
    Book(int id, double price, int number);
    //get
    int get_id();//get book id
    int get_number();//get book stock
    double get_price();//get book price
    //set
    void set_number(int n_number);//set book stock
    void set_id(int id);//set book id
    void set_price(double price);//set book price
    void show(int id);//show book information
private:
    int number;//book stock
    int  id;//book id
    double price;//book price

};
```

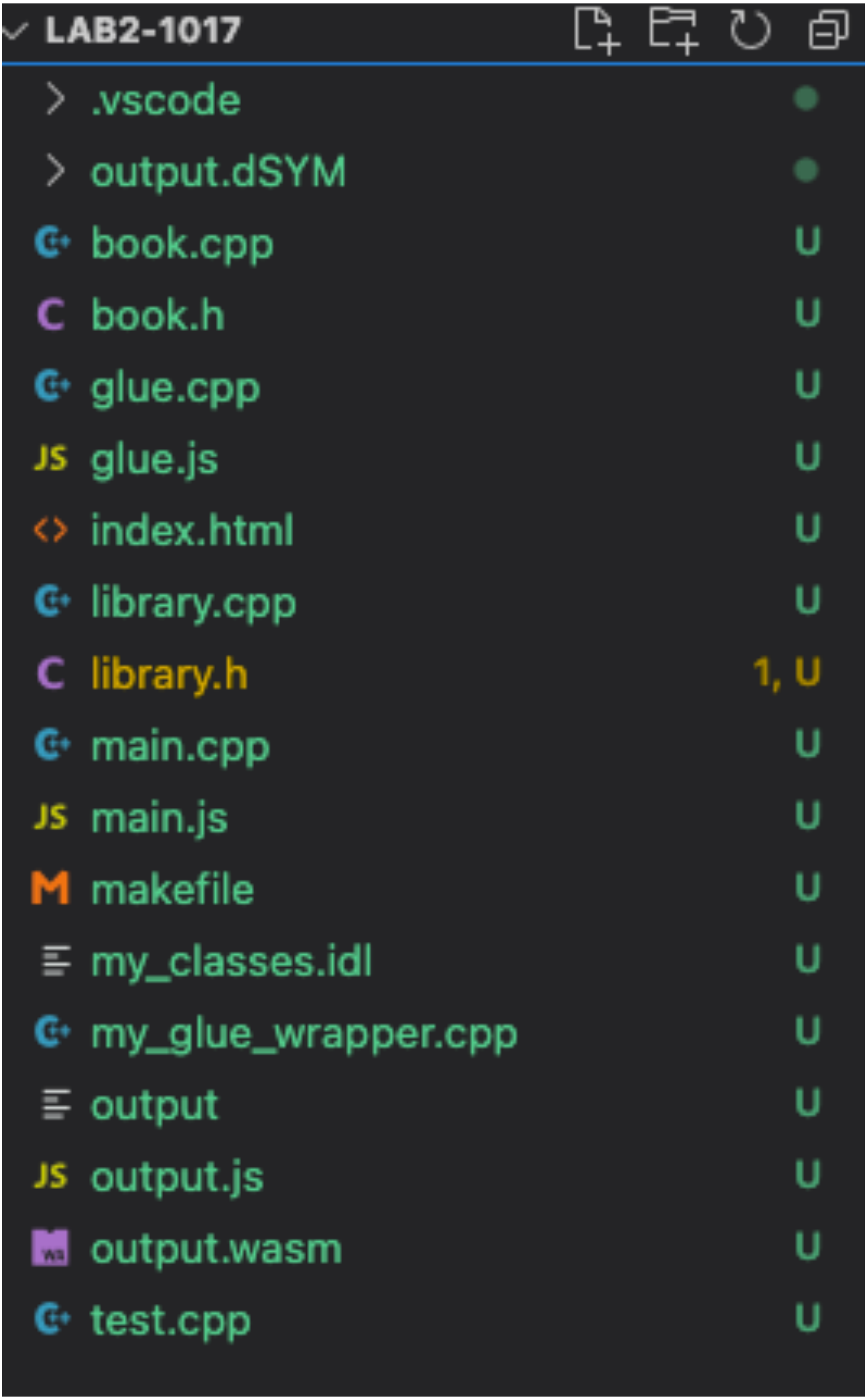
library.h

```
class Library
{
```

```
public:
    Library();
    int get_m();//get m
    bool search();//search book
    void set_book1(int id, double price, int 
number);//create a new book
    bool borrow_book(int id, int num);//borrow a book
    void show();//show library information
    double sale_sum();//calculate total sales
    double sale_average();//calculate average sales
    void sale_show(double sum, double average);//show 
sales information

private:
    int m;//total book number
    int sale_num=0;//total sale number
    Book book1[1000];
    Book book_borrow[1000];
};
```

b. Provide a second section that describes the design concepts on how the library was created. Be sure to describe in detail any design patterns, class structure and provide explanations, pro/cons for design decisions and implementations.



LAB2-1017

> .vscode	●
> output.dSYM	●
book.cpp	U
book.h	U
glue.cpp	U
glue.js	U
index.html	U
library.cpp	U
library.h	1, U
main.cpp	U
main.js	U
makefile	U
my_classes.idl	U
my_glue_wrapper.cpp	U
output	U
output.js	U
output.wasm	U
test.cpp	U

book.h and book.cpp are used to implement book related methods: create books according to basic information (ID, price, number), get / set book number, price ID;

library.h and library.cpp are used to implement library related methods such as creating get/set total book number, showing book information, calculating total sales and average sales. In the library.cpp, borrow_book() function will first search if there are the book that will be borrowed and if there are enough book to be borrowed. When a student borrows n books, the stock of this book will be reduced by n. If there are not enough books for students to borrow, the system will return a prompt.

test.cpp realizes the process of borrowing books and calculating sales volume in the form of C ++ and main.js realizes the process of borrowing books and calculating sales volume in the form of JavaScript.

my_classes.idl is the webIDL. This file includes the interfaces of all public methods.

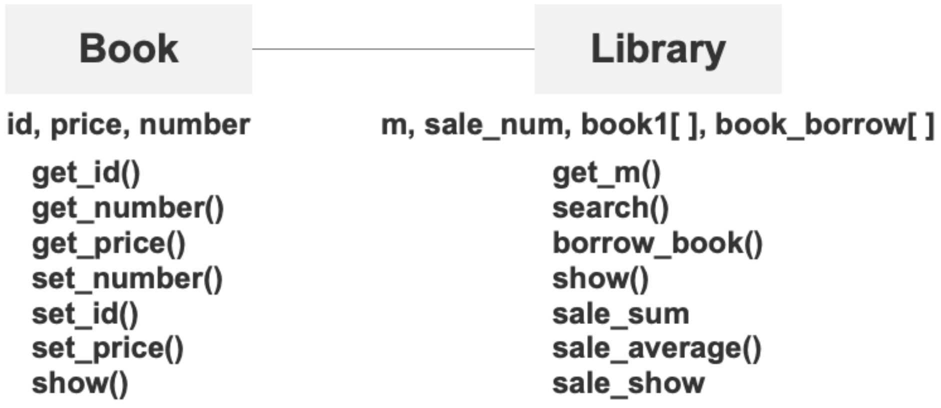
my_glue_wrapper.cpp includes all .h file of class and glue.cpp.

glue.cpp and glue.js are created by 'python tools/webidl_binder.py my_classes.idl glue' under emscripten.

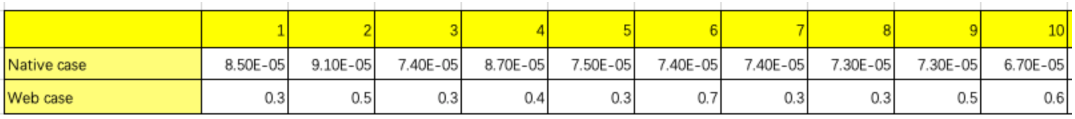
output.js is a JS format file that C + + programs are packaged into. By reference, we can call the encapsulated method.

main.js is the logical implementation of web pages; index.js is the page construction of web pages.

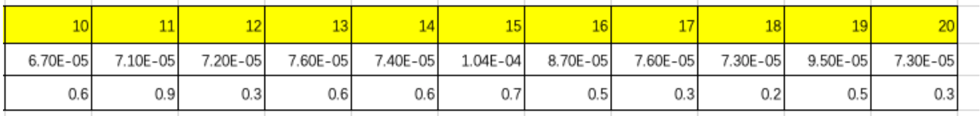
c. High-level class layout/UML should be included in writeup



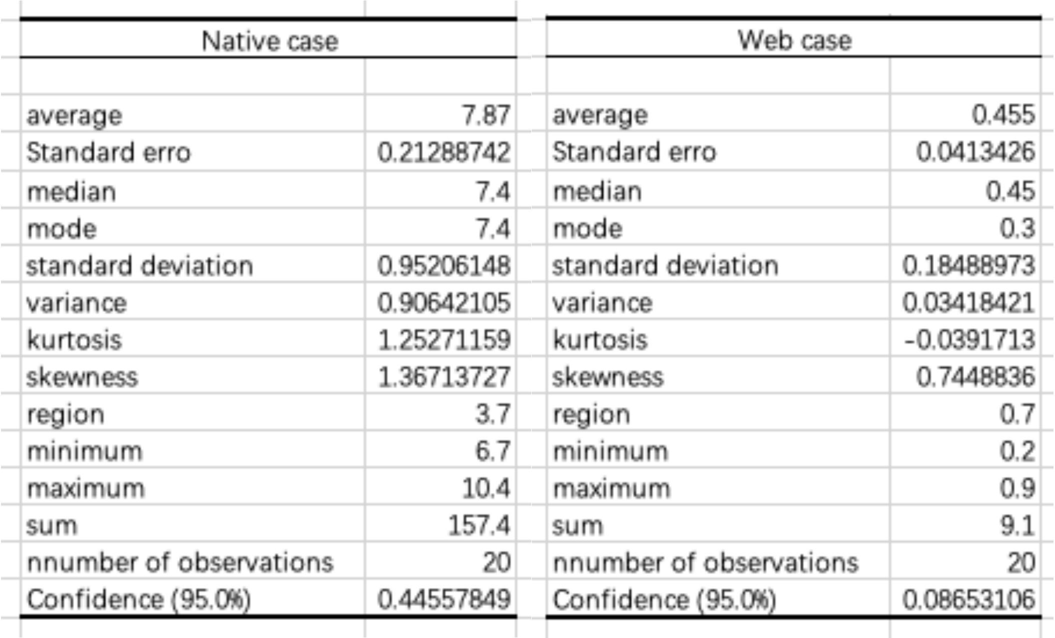
2. Compare and Contrast execution time of the library code between demo application and comparison application



	1	2	3	4	5	6	7	8	9	10
Native case	8.50E-05	9.10E-05	7.40E-05	8.70E-05	7.50E-05	7.40E-05	7.40E-05	7.30E-05	7.30E-05	6.70E-05
Web case	0.3	0.5	0.3	0.4	0.3	0.7	0.3	0.3	0.5	0.6



10	11	12	13	14	15	16	17	18	19	20
6.70E-05	7.10E-05	7.20E-05	7.60E-05	7.40E-05	1.04E-04	8.70E-05	7.60E-05	7.30E-05	9.50E-05	7.30E-05
0.6	0.9	0.3	0.6	0.6	0.7	0.5	0.3	0.2	0.5	0.3



Native case		Web case	
average	7.87	average	0.455
Standard erro	0.21288742	Standard erro	0.0413426
median	7.4	median	0.45
mode	7.4	mode	0.3
standard deviation	0.95206148	standard deviation	0.18488973
variance	0.90642105	variance	0.03418421
kurtosis	1.25271159	kurtosis	-0.0391713
skewness	1.36713727	skewness	0.7448836
region	3.7	region	0.7
minimum	6.7	minimum	0.2
maximum	10.4	maximum	0.9
sum	157.4	sum	9.1
nnumber of observations	20	nnumber of observations	20
Confidence (95.0%)	0.44557849	Confidence (95.0%)	0.08653106

use Visual Studio Code to run make file to run the local programs, and Visual Studio Code and live server to run web programs.

It can be seen from the above data that the running speed of the local program is obviously faster than that of the web end, and it is more stable than that of the web end.