

C++ Programming	Student number	21600193
Homework 4	Name	Kim, Hyo Rim

1. Comment

1) struct book{}: Make the information of the book to be saved into struct format. 'title', 'p_Year', 'author', 'edition', 'b_Person' are declared as string type. Declare 'l_Day' and 'index' as int. At this time, 'b_Person' and 'l_Day' specify default values. int index represents the number of stored books.

2) int main(): Declare b_List, the pointer type of struct book. Use the 'load_m' function to store the contents of the file in b_List. Use the while statement to continue the menu until the 'exit' command is called. Each command uses a switch to call the required function. In this case, the command uses the toupper function so that it is not case sensitive.

3) void printMenu(): Show the menu.

4) string * p_Arg(string line): Divide the input command so that it can be used in the program and save it.

5) book *load_m(string FileName): It receives the file name and stores the file information in a struct book. At this point, struct book uses static variables to make the saved book available to other functions as well.

6) void save_m(string fileName, book * bookList): Save the struct book as a file name.

7) void print_m(book * bookList): Receives a struct book and prints out the Book Catalog and a statement specifying the return time limit for the book.

8) book * insert_m(book * bookList, string * arg): The new book information inputted from the user through the command is stored in the existing information and outputted.

9) book *lend_m(book * bookList, string * arg): It stores information of a book borrowed by another person.

10) book *passDay_m(book * bookList): Subtract 1 from 'l_Day' of the book where 'b_Person' is not "None".

11) book * returned_m(book * bookList, string * arg): When the returned book is entered, the 'l_Day' and 'b_Person' of the book are returned to their default values.

2. Code

```

#include <iostream>
#include <string>
#include<iomanip>
#include <fstream>

using namespace std;

//DONE: ONE - make
struct book{

    string title;
    string p_Year;
    string author;
    string edition;
    string b_Person = "None";
    int l_Day = 0;
    int index;

};

void printMenu();
string * p_Arg(string line);
book *load_m(string FileName);
void save_m(string fileName, book * bookList);
void print_m(book * bookList);
book * insert_m(book * bookList, string * arg);
book *lend_m(book * bookList, string * arg);
book *passDay_m(book * bookList);
book * returned_m(book * bookList, string * arg);

int getExit();

int main(){

```

```

int setExit = 1;

book * b_List;

string openFile = "input.txt";

b_List = load_m(openFile);

while(setExit){

    string argLine, * arg;

    printMenu();

    getline(cin, argLine);

    arg = p_Arg(argLine);

    char n1 = arg[0][0];
    char n2 = arg[0][1];

    switch(toupper(n1)){

    case 'I':

        b_List = insert_m(b_List, arg);
        break;

    case 'L' :

        b_List = lend_m(b_List, arg);
        break;

    case 'S':

        save_m(arg[1], b_List);
        break;

    case 'R':

        b_List = returned_m(b_List, arg);

```

```

                                break;

        case 'P':

            switch (toupper(n2)) {

                case 'A':

                    b_List = passDay_m(b_List);
                    break;

                case 'R':

                    print_m(b_List);
                    break;

            }
            break;

        case 'E':

            setExit = getExit();
            break;

    }

}

return 0;
}

void printMenu(){

    cout << " ===== Available Commend & Format =====" << endl << endl;
    cout << " 1. INSERT BookTitle; Author; PubYear; Edition" << endl;
    cout << " 2. LEND BookTitle; Person Borrowing; How many days" << endl;
    cout << " 3. SAVE new_filename.txt" << endl;
    cout << " 4. RETURNED BookTitle" << endl;
    cout << " 5. PASSDAY" << endl;
    cout << " 6. PRINT" << endl;
    cout << " 7. EXIT" << endl << endl;
    cout << " =====" << endl;
    cout << " >>";
}

```

```

}

string * p_Arg(string line){

    int f_col, i = 0;

    static string arg[6];

    while(1){

        if (i == 0) {
            f_col = line.find(" ");
            arg[i] = line.substr(0, f_col);
            line.erase(0, f_col + 1);
            i++;
        }
        else {
            if (line.length() > 0) {
                f_col = line.find(";");
                if (f_col == -1) {
                    arg[i] = line;
                    break;
                }
                arg[i] = line.substr(0, f_col);
                line.erase(0, f_col + 2);
                i++;
            }
            else
                break;
        }
    }

    return arg;
}

```

```

}

//DONE: THREE - function Load

book *load_m(string FileName){
    //open file and store list
    static book bookList[50];

    int i = 0;

    string line;

    ifstream inData;
    inData.open(FileName.c_str());

    if(!inData){

        cout << FileName << " does not exist." << endl;

    }
    else{

        while(!inData.eof()){

            getline(inData, line);

            int f_col, j = 0;
            string arr[6];

            while(1){

                if (line.length() > 0) {

```

```

        f_col = line.find(";");
        if (f_col == -1) {
            arr[j] = line;
            break;
        }
        arr[j] = line.substr(0, f_col);
        line.erase(0, f_col+2);
        j++;
    }
    else
        break;
}

bookList[i].title = arr[0];
bookList[i].p_Year = arr[1];
bookList[i].author = arr[2];
bookList[i].edition = arr[3];
bookList[i].b_Person = arr[4];
bookList[i].l_Day = stoi(arr[5]);

i++;

}
i--;
bookList[0].index = i;
print_m(bookList);
}

inData.close();

return bookList;

```

```

}

//DONE: THREE - function Save

void save_m(string fileName, book * bookList){
    // use ofstream and save text file
    ofstream outData;

    int i = 0;

    outData.open(fileName.c_str());

    while (i < bookList[0].index - 1) {

        outData << bookList[i].title << " "; " << bookList[i].p_Year << " "; " <<
bookList[i].author << " "; "<< bookList[i].edition << " "; "<< bookList[i].b_Person << " "; "<<
bookList[i].l_Day << endl;

        i++;

    }

    outData << bookList[i].title << " "; "<< bookList[i].p_Year << " "; "<< bookList[i].author
<< " "; "<< bookList[i].edition << " "; " << bookList[i].b_Person << " "; "<< bookList[i].l_Day;

    outData.close();
}

//DONE: THREE - function Print

void print_m(book * bookList){

    string pass[50];
    string today[50];
    int pass_j = 0, today_j = 0;

```



```

        cout << endl << "      ===== Book Catalog
===== " << endl;

        cout << setw(8) << "Title" << setw(24) << "Author" << setw(25) << "Publised Year" <<
setw(10) << "Edition" << setw(10) << "Borrower" << setw(16) << "Days Borrowed" << endl;

        for(int i = 0; i < bookList[0].index ; i++ ){
            if(bookList[i].b_Person != "None"){
                if(bookList[i].l_Day == 0){
                    today[today_j] = bookList[i].title;
                    today[today_j+1] = bookList[i].b_Person;
                    today_j += 2;
                }
                else if(bookList[i].l_Day <0){
                    pass[pass_j] = bookList[i].title;
                    pass[pass_j+1] = bookList[i].b_Person;
                    pass_j += 2;
                }
            }

            cout << "      " << bookList[i].title << setw(23 - bookList[i].title.length() +
bookList[i].author.length()) << bookList[i].author << setw(22 - bookList[i].author.length()) <<
bookList[i].p_Year << setw(22 - bookList[i].p_Year.length()) << bookList[i].edition << setw(9
- bookList[i].edition.length() + bookList[i].b_Person.length()) << bookList[i].b_Person <<
setw(12 - bookList[i].b_Person.length()) << bookList[i].l_Day << endl;
        }

        cout << "      ===== End
===== " << endl;

        for(int i = 0; i < pass_j ; i += 2)
            cout << pass[i]<< " SHOULD HAVE BEEN RETURNED ALREADY by " << pass[i+1]<< endl;

        for(int i = 0; i < today_j ; i += 2)
            cout << today[i]<< " should be returned today by " << today[i+1]<< endl;

```

```

        cout << endl;
    }

//DONE: THREE - function Insert
book * insert_m(book * bookList, string * arg){

    bookList[bookList[0].index].title = arg[1];
    bookList[bookList[0].index].author = arg[2];
    bookList[bookList[0].index].p_Year = arg[3];
    bookList[bookList[0].index].edition = arg[4];

    bookList[0].index++;

    cout << endl << "  " << "Inserted " << arg[1] << " successfully!" << endl << endl;

    print_m(bookList);

    return bookList;

}

//DONE: THREE - function Lend
book *lend_m(book * bookList, string * arg){

    int j = 0;

    for(int i = 0; i < bookList[0].index; i++){

        if(bookList[i].title == arg[1])
            j++;

    }

    if(j == 0){

```

```

        cout << "NO SUCH BOOK!" << endl;

        return bookList;

    }

    cout << endl << "      ===== Books Cuurrently Lent
===== " << endl;

    cout << setw(8) << "Title" << setw(24) << "Author" << setw(25) << "Publised Year" <<
    setw(10) << "Edition" << setw(10) << "Borrower" << setw(16) << "Days Borrowed" << endl;

    for(int i = 0; i < bookList[0].index; i++){

        if(bookList[i].title == arg[1]){

            bookList[i].b_Person = arg[2];
            bookList[i].l_Day = stoi(arg[3]);
        }

        if(bookList[i].b_Person != "None")
            cout << "      " << bookList[i].title << setw(23 -
bookList[i].title.length() + bookList[i].author.length()) << bookList[i].author << setw(22 -
bookList[i].author.length()) << bookList[i].p_Year << setw(22 - bookList[i].p_Year.length()) <<
bookList[i].edition << setw(9 - bookList[i].edition.length() + bookList[i].b_Person.length())
<< bookList[i].b_Person << setw(12 - bookList[i].b_Person.length()) << bookList[i].l_Day <<
endl;

    }

    cout << "      ===== End
===== " << endl << endl;

    return bookList;
}

//DONE: THREE - function Passday

```

```

book *passDay_m(book * bookList){

    cout << endl <<"      ===== Books Cuurrently Lent
===== " <<endl;

    cout << setw(8) << "Title" << setw(24) << "Author" << setw(25) << "Publised Year" <<
setw(10) << "Edition" << setw(10) << "Borrower" << setw(16) << "Days Borrowed" << endl;

    string pass[50];
    string today[50];
    int pass_j = 0, today_j = 0;

    for(int i = 0; i < bookList[0].index; i++){

        if(bookList[i].b_Person != "None"){

            bookList[i].l_Day--;

            if(bookList[i].l_Day == 0){
                today[today_j] = bookList[i].title;
                today[today_j+1] = bookList[i].b_Person;
                today_j += 2;
            }
            else if(bookList[i].l_Day <0){
                pass[pass_j] = bookList[i].title;
                pass[pass_j+1] = bookList[i].b_Person;
                pass_j += 2;
            }

            cout << "      " << bookList[i].title << setw(23 -
bookList[i].title.length() + bookList[i].author.length()) << bookList[i].author << setw(22 -
bookList[i].author.length()) << bookList[i].p_Year << setw(22 - bookList[i].p_Year.length()) <<
bookList[i].edition << setw(9 - bookList[i].edition.length() + bookList[i].b_Person.length())
<< bookList[i].b_Person << setw(12 - bookList[i].b_Person.length()) << bookList[i].l_Day <<
endl;

```

```

        }
    }

    cout << "===== End
===== " << endl;

    for(int i = 0; i < pass_j ; i += 2)
        cout << pass[i]<< " SHOULD HAVE BEEN RETURNED ALREADY by " << pass[i+1]<< endl;

    for(int i = 0; i < today_j ; i += 2)
        cout << today[i]<< " should be returned today by " << today[i+1]<< endl;
    cout << endl;

    return bookList;
}

//DONE: THREE - function Returned
book * returned_m(book * bookList, string * arg){

    int j = 0;

    for(int i = 0; i < bookList[0].index; i++){

        if(bookList[i].title == arg[1]){
            if(bookList[i].b_Person == "None"){
                cout << endl<< "ATTENTION No one borrowed that book!!" << endl;
                return bookList;
            }
            else{
                j++;
                cout << endl << "===== Book
Returned =====" <<endl;

                cout << setw(8) << "Title" << setw(24) << "Author" << setw(25) <<

```

```

"Publised Year" << setw(10) << "Edition" << setw(10) << "Borrower" << setw(16) << "Days Borrowed"
<< endl;

        bookList[i].b_Person = "None";
        bookList[i].l_Day = 0;

        cout << "          " << bookList[i].title << setw(23 -
bookList[i].title.length() + bookList[i].author.length()) << bookList[i].author << setw(22 -
bookList[i].author.length()) << bookList[i].p_Year << setw(22 - bookList[i].p_Year.length()) <<
bookList[i].edition << setw(9 - bookList[i].edition.length() + bookList[i].b_Person.length())
<< bookList[i].b_Person << setw(12 - bookList[i].b_Person.length()) << bookList[i].l_Day <<
endl;

        cout << "          ===== End
===== " << endl << endl;

    }

}

}

if(j == 0)
    cout << endl << "ATTENTION No Such Book!" << endl;

return bookList;
}

//DONE: THREE - fundtion Exit
int getExit(){

    return 0;

}

```

3. Result

1)input.txt

2)Start & Insert

```
hyorm@Hyorm:/mnt/c/Users/Hyo RIM Kim/Documents/한동대학교/Git/17-2/C++$ ./hw4
===== Book Catalog =====
Title Author Published Year Edition Borrower Days Borrowed
Discrete Mathematics Douglas Winston 1998 5th Ed None 0
C++ Neil Dale 2016 6th Ed None 0
How To Program Daniel Craig 2017 1st Ed None 0
===== End =====

===== Available Command & Format =====
1. INSERT BookTitle; Author; PubYear; Edition
2. LEND BookTitle; Person Borrowing; How many days
3. SAVE new_filename.txt (core dumped)
4. RETURNED BookTitle
5. PASSDAY is the significance of segmentation fault (core dumped). I searched and it seems an error
6. PRINT that occurs when you try to access unallocated memory, so, what's wrong with the above
7. EXIT code?

=====
>>inSeRt The Hovvit; Tolkien; 1980; 1st Ed
Inserted The Hovvit successfully!

===== Book Catalog =====
Title Author Published Year Edition Borrower Days Borrowed
Discrete Mathematics Douglas Winston 1998 5th Ed None 0
C++ Neil Dale 2016 6th Ed None 0
How To Program Daniel Craig 2017 1st Ed None 0
The Hovvit Tolkien 1980 1st Ed None 0
===== End =====

===== Available Command & Format =====
1. INSERT BookTitle; Author; PubYear; Edition
2. LEND BookTitle; Person Borrowing; How many days
3. SAVE new_filename.txt i++) //0,1,2,3 = total 4 values
4. RETURNED BookTitle
5. PASSDAY cout << str[i] << "\n";
6. PRINT
7. EXIT

>>
```

3)Returned

```
4. RETURNED BookTitle
5. PASSDAY is the significance of segmentation fault (core dumped). I searched
6. PRINT that occurs when you try to access unallocated memory, so, what's
7. EXIT code?

=====
>>Returned C++
ATTENTION No one borrowed that book!!

===== Available Command & Format =====
1. INSERT BookTitle; Author; PubYear; Edition
2. LEND BookTitle; Person Borrowing; How many days
3. SAVE new_filename.txt
4. RETURNED BookTitle
5. PASSDAY
6. PRINT
7. EXIT

=====
>>Returned C
you should write:
ATTENTION No Such Book!

===== Available Command & Format =====
1. INSERT BookTitle; Author; PubYear; Edition
2. LEND BookTitle; Person Borrowing; How many days
3. SAVE new_filename.txt
```

4) Lend

```

6. PRINT
7. EXIT
=====
>>lend C++; Friend; 3
=====
Books Cuurrently Lent
=====
Title Author Publied Year Edition Borrower Days Borrowed
C++ What is the significance of segmentation fault (core dumped). I searched and it seems an error like that occurs when you try to access an array. End of the world, so, what's wrong with the above code?
=====
Available Command & Format
=====
1. INSERT BookTitle; Author; PubYear; Edition
2. LEND BookTitle; Person Borrowing; How many days
3. SAVE new_filename.txt
4. RETURNED BookTitle
5. PASSDAY
6. PRINT
7. EXIT
=====
Use g++ -Wall -g to compile and debug with gdb - Basile Starynkevitch Jun 24 '12 at 13:45
=====
>>lend Discrete Mathematics; girlfriend; 1
=====
Books Cuurrently Lent
=====
Title Author Publied Year Edition Borrower Days Borrowed
Discrete Mathematics Douglas Winston 1998 5th Ed girlfriend 1
C++ What is the significance of segmentation fault (core dumped). I searched and it seems an error like that occurs when you try to access an array. End of the world, so, what's wrong with the above code?
=====
Available Command & Format
=====
1. INSERT BookTitle; Author; PubYear; Edition
2. LEND BookTitle; Person Borrowing; How many days

```

5) Passday

```

6. PRINT
7. EXIT
=====
>>passday
=====
Books Cuurrently Lent
=====
Title Author Publied Year Edition Borrower Days Borrowed
Discrete Mathematics Douglas Winston 1998 5th Ed girlfriend-1
C++ What is the significance of segmentation fault (core dumped). I searched and it seems an error like that occurs when you try to access an array. End of the world, so, what's wrong with the above code?
=====
Available Command & Format
=====
1. INSERT BookTitle; Author; PubYear; Edition
2. LEND BookTitle; Person Borrowing; How many days
3. SAVE new_filename.txt
4. RETURNED BookTitle
5. PASSDAY
6. PRINT
7. EXIT
=====
Use g++ -Wall -g to compile and debug with gdb - Basile Starynkevitch Jun 24 '12 at 13:45
=====
>>passday
=====
Books Cuurrently Lent
=====
Title Author Publied Year Edition Borrower Days Borrowed
Discrete Mathematics Douglas Winston 1998 5th Ed girlfriend-2
C++ What is the significance of segmentation fault (core dumped). I searched and it seems an error like that occurs when you try to access an array. End of the world, so, what's wrong with the above code?
=====
Available Command & Format
=====
1. INSERT BookTitle; Author; PubYear; Edition

```


6)Print

```

6. PRINT
7. EXIT

=====
>>print
=====
Book Catalog
=====
Title          Author      Publied Year  Edition  Borrower  Days Borrowed
Discrete Mathematics  Douglas Winston  1998      5th Ed  girlfriend-2
C++            Neil Dale    2016      6th Ed  Friend    0
HowTo Program  Daniel Craig  2017      1st Ed  None      0
The Hovvit     Tolkien      1980      1st Ed  None      0
=====
Discrete Mathematics SHOULD HAVE BEEN RETURNED ALREADY by girlfriend
C++ should be returned today by Friend

===== Available Command & Format =====
1. INSERT BookTitle; Author; PubYear; Edition
2. LEND BookTitle; Person Borrowing; How many days

```

7)Returned correctly

```

5. PASSDAY
6. PRINT
7. EXIT

=====
>>returned Discrete Mathematics
6 Answers
=====
Book Retured
=====
Title          Author      Publied Year  Edition  Borrower  Days Borrowed
Discrete Mathematics  Douglas Winston  1998      5th Ed  None      0
=====
===== Available Command & Format =====
1. INSERT BookTitle; Author; PubYear; Edition
2. LEND BookTitle; Person Borrowing; How many days

```

8) Save

```

7. EXIT

=====
>>SAVE output.txt
===== Available Command & Format =====
1. INSERT BookTitle; Author; PubYear; Edition
2. LEND BookTitle; Person Borrowing; How many days
3. SAVE new_filename.txt
4. RETURNED BookTitle
5. PASSDAY
6. PRINT
7. EXIT

for (int i = 0; i<4; i++) //0,1,2,3 = total values
=====
>>exit
cout << str[i] << "\n";

Hyorm@HYorm: /mnt/c/Users/Hyo_RIM_Kim/Documents/한동대학교/Git/17-2/C++$ ls
HW4_21600193_KimHyoRim.cpp  hw4      output.txt
HW4_21600193_KimHyoRim.docx  input.txt  ~$4_21600193_KimHyoRim.docx
Hyorm@HYorm: /mnt/c/Users/Hyo_RIM_Kim/Documents/한동대학교/Git/17-2/C++$ cat output.txt
Discrete Mathematics; 1998; Douglas Winston; 5th Ed; None; 0
C++; 2016; Neil Dale; 6th Ed; Friend; 0
How To Program; 2017; Daniel Craig; 1st Ed; None; 0
The Hovvit; 1980; Tolkien; 1st Ed; None; 0
Hyorm@HYorm: /mnt/c/Users/Hyo_RIM_Kim/Documents/한동대학교/Git/17-2/C++$

```