CS 201, Summer 2022
Homework Assignment 1

Due: 23:55, June 27, 2022

Important Notes:

Please do not start the assignment before reading these notes.

1. This assignment is due by 23:55, June 27. You should upload your homework to the upload link on Moodle before the deadline. This upload link will be available between June 18 and June 28. The standard rules about late homework submissions apply. Please see the course syllabus for further discussion of the late homework policy as well as academic integrity.

2. Your code must use dynamically allocated arrays using pointers to store book and student data. You will get no points if you use static arrays with fixed sizes or other data structures such as `vector` from the standard library.

3. Your code must not have any memory leaks. You will lose points if you have memory leaks in your program even though the outputs of the operations are correct.

4. In this assignment, you must have separate interface and implementation files (i.e., separate `.h` and `.cpp` files) for your class. We will test your implementation by writing our own driver `.cpp` file which will include your header file. For this reason, your class' name MUST BE "`LibrarySystem`" and your files' name MUST BE "`LibrarySystem.h`" and "`LibrarySystem.cpp`". You should upload these two files (and any additional files if you wrote additional classes in your solution) as a single archive file (e.g., zip, tar, rar). The submissions that do not obey these rules will not be graded.

   We also recommend you to write your own driver file to test each of your functions. However, you MUST NOT submit this test code (we will use our own test code). In other words, your submitted code should not include any `main` function.

5. You are free to write your programs in any environment (you may use either Linux or Windows). On the other hand, we will test your programs on "`dijkstra.ug.bcc.bilkent.edu.tr`" and we will expect your programs to compile and run on the "`dijkstra`" machine. If we could not get your program properly work on the "`dijkstra`" machine, you would lose a considerable amount of points. Therefore, we recommend you to make sure that your program compiles and properly works on "`dijkstra.ug.bcc.bilkent.edu.tr`" before submitting your assignment.

6. This homework will be graded by your TA Mert Kara (mert.kara at bilkent.edu.tr). Thus, you may ask your homework related questions directly to him.

In this homework, you will implement a library system. The library system will maintain a collection of books for the library. The students will be able to check out books from the library and will be able to return them to the library.

The library system will have the following functionalities; the details of these functionalities are given below:

1. Add a book
2. Delete a book
3. Add a student
4. Delete a student
5. Check out a book by a student
6. Return a book

7. Show the list of all books
8. Show detailed information about a particular book
9. Show detailed information about a particular student

Add a book: The library system will allow to add a new book to its collection with the following information: book id, book title, authors, and the year it was published. The book ids must be unique positive integers. Hence, the system should check whether or not the specified book id already exists (i.e., whether or not it is the id of another book), and if the book exists, it should not allow the operation and display a warning message.

Delete a book: The library system will allow to delete an existing book specified with its book id. If the book does not exist (i.e., if there is no book with the specified id), the system should display a warning message. Note that it should be possible to delete books which are checked out.

Add a student: The library system will allow to add a new student with the following information: student id and name. The student ids must be unique positive integers. Therefore, the system should check whether or not the specified student id already exists (i.e., whether or not it is the id of another student), and if the student exists, it should not allow the operation and display a warning message.

Delete a student: The library system will allow to delete an existing student with student id. If the student does not exist (i.e., if there is no student with the specified id), the system should display a warning message. If the student has checked out books, those books must be returned to the system as part of the delete operation for the student.

Checkout a book by a student: The library system will allow to check out a particular book by a particular student. For that, the book id and the student id have to be specified. The system should first check whether or not this book exists; if it does not, it should not allow the operation and display a warning message. The system should also check whether or not this student exists; if he/she does not, it should not allow the operation and display a warning message. If both the book and the student exist and the book is not already checked out, the check out operation must be performed. Note that a book can be checked out by only one student. However, a student can check out multiple books.

Return a book: The library system will allow a student to return a book. For that, the book id has to be specified. If the book does not exist, the system will display a warning message. It will also check whether or not this book is checked out; if not, a warning message will be displayed, otherwise, the operation will be carried out.

Show the list of all books: The library system will allow to display the list of all the books. This list includes, for each book, the book id, book title, authors, and publication year. For each book that is checked out, id and name of the student who checked out the book should also be displayed.

Show detailed information about a particular book: The library system will allow to specify a book id and display detailed information about that particular book. This information includes the book id, book title, authors, publication year and whether or not the book was checked out. If the book is checked out, id and name of the student who checked out the book should be displayed. If the book does not exist (i.e., if there is no book with the specified book id), the system should display a warning message.

Show detailed information about a particular student: The library system will allow to specify a student id and display detailed information about that particular student. This information includes the student id, student name, and the list of books checked out by this student including the book id, book title, authors, and publication year. If the student does not exist (i.e., if there is no student with the specified student id), the system should display a warning message.

Below is the required `public` part of the `LibrarySystem` class that you must write in this assignment. The name of the class must be `LibrarySystem`, and must include these public member functions. We will use these functions to test your code. The interface for the class must be written in a file called `LibrarySystem.h` and its implementation must be written in a file called `LibrarySystem.cpp`. You can define additional public and private member functions and data members in this class. You can also define additional classes in your solution.

```
class LibrarySystem {

public:
    LibrarySystem();
    ~LibrarySystem();

    void addBook(const int bookId, const string name, const string authors,
        const int year);
    void deleteBook(const int bookId);

    void addStudent(const int studentId, const string name);
    void deleteStudent(const int studentId);

    void checkoutBook(const int bookId, const int studentId);
    void returnBook(const int bookId);

    void showAllBooks();
    void showBook(const int bookId);
    void showStudent(const int studentId);
};
```

Here is an example test program that uses this class and the corresponding output. We will use a similar program to test your solution so make sure that the name of the class is `LibrarySystem`, its interface is in the file called `LibrarySystem.h`, and the required functions are defined as shown above. You can assume that the book ids and student ids are entered as positive integers.

Example test code:

```
#include "LibrarySystem.h"

int main() {

    LibrarySystem LS;

    LS.addBook(1000, "Machine Learning", "Tom Mitchell", 1997);
    LS.addBook(1200, "Data Mining", "Michael S., Vipin K.", 1991);
    LS.addBook(1300, "Problem S. with C++", "Frank M. Carrano", 1991);
    LS.addBook(1400, "C++ How to Program", "Deitel & Deitel", 2005);
    LS.addBook(1200, "Data Mining", "Michael S., Vipin K.", 1991);
    LS.deleteBook(1300);
    LS.deleteBook(2000);
    LS.addBook(1500, "Pattern Recognition", "Duda, Hart, Stork", 2000);
    cout << endl;
    LS.addStudent(21000000, "Ali Tekin");
    LS.addStudent(21000011, "Hasan Ak");
```

```
    LS.addStudent(21000011, "Fadime Sener");
    LS.addStudent(21000020, "Gokhan Karaca");
    LS.addStudent(21000001, "Eren Tunc");
    LS.addStudent(21000005, "Merve Zarif");
    LS.deleteStudent(21000011);
    LS.deleteStudent(21000050);
    cout << endl;
    LS.checkoutBook(1200, 21000000);
    LS.checkoutBook(1400, 21000020);
    LS.checkoutBook(2050, 21000011);
    LS.checkoutBook(1000, 21000444);
    LS.checkoutBook(1500, 21000000);
    LS.checkoutBook(1400, 21000001);
    cout << endl;
    LS.showStudent(21000000);
    cout << endl;
    LS.showStudent(21000005);
    cout << endl;
    LS.showStudent(21000011);
    cout << endl;
    LS.showBook(1000);
    LS.showBook(1200);
    cout << endl;
    LS.showAllBooks();
    cout << endl;
    LS.returnBook(1200);
    LS.returnBook(1000);
    cout << endl;
    LS.checkoutBook(1200, 21000020);
    LS.checkoutBook(1000, 21000000);
    cout << endl;
    LS.showAllBooks();
    cout << endl;
    LS.deleteStudent(21000020);
    cout << endl;
    LS.deleteBook(1000);
    cout << endl;
    LS.showStudent(21000000);
    cout << endl;
    LS.showAllBooks();

    return 0;
}
```

Output of the example test code:

```
Book 1000 has been added
Book 1200 has been added
Book 1300 has been added
Book 1400 has been added
Book 1200 already exists
```

```
Book 1300 has not been checked out
Book 1300 has been deleted
Book 2000 does not exist
Book 1500 has been added

Student 21000000 has been added
Student 21000011 has been added
Student 21000011 already exists
Student 21000020 has been added
Student 21000001 has been added
Student 21000005 has been added
Student 21000011 has been deleted
Student 21000050 does not exist

Book 1200 has been checked out by student 21000000
Book 1400 has been checked out by student 21000020
Book 2050 does not exist for checkout
Student 21000444 does not exist for checkout
Book 1500 has been checked out by student 21000000
Book 1400 has been already checked out by another student

Student id: 21000000 student name: Ali Tekin
Student 21000000 has checked out the following books:
Book id   Book name              Authors                 Year
1200      Data Mining            Michael S., Vipin K.    1991
1500      Pattern Recognition    Duda, Hart, Stork       2000

Student id: 21000005 student name: Merve Zarif
Student 21000005 has no books

Student 21000011 does not exist

1000      Machine Learning       Tom Mitchell            1997      Not checked out
1200      Data Mining            Michael S., Vipin K.    1991      Checked out by student 21000000

Book id   Book name              Authors                 Year      Status
1000      Machine Learning       Tom Mitchell            1997      Not checked out
1200      Data Mining            Michael S., Vipin K.    1991      Checked out by student 21000000
1400      C++ How to Program     Deitel & Deitel         2005      Checked out by student 21000020
1500      Pattern Recognition    Duda, Hart, Stork       2000      Checked out by student 21000000

Book 1200 has been returned
Book 1000 has not been checked out

Book 1200 has been checked out by student 21000020
Book 1000 has been checked out by student 21000000

Book id   Book name              Authors                 Year      Status
1000      Machine Learning       Tom Mitchell            1997      Checked out by student 21000000
1200      Data Mining            Michael S., Vipin K.    1991      Checked out by student 21000020
1400      C++ How to Program     Deitel & Deitel         2005      Checked out by student 21000020
1500      Pattern Recognition    Duda, Hart, Stork       2000      Checked out by student 21000000

Book 1400 has been returned
Book 1200 has been returned
Student 21000020 has been deleted
```

```
Book 1000 has been returned
Book 1000 has been deleted

Student id: 21000000 student name: Ali Tekin
Student 21000000 has checked out the following books:
Book id   Book name              Authors                Year
1500      Pattern Recognition    Duda, Hart, Stork      2000

Book id   Book name              Authors                Year      Status
1200      Data Mining            Michael S., Vipin K.   1991      Not checked out
1400      C++ How to Program     Deitel & Deitel        2005      Not checked out
1500      Pattern Recognition    Duda, Hart, Stork      2000      Checked out by student 21000000
```