CHANDIGARH UNIVERSITY

Discover. Learn. Empower.

UNIVERSITY INSTITUTE OF COMPUTING

PROJECT ON

LIBRARY MANAGEMENT SYSTEM

Program name:- BCA

SUBJECT NAME/CODE: COMPUTER APTITUTE/ 23CAP-308

Submitted by:-

Name: Aditya

UID: 23BCA10549

Section : 23BCA-9

Group: B

Submitted to:-

Name: Mrs.PriyankaArora

Acknowledgement

I would like to express my sincere gratitude to Chandigarh University for providing me with the opportunity to undertake this mini project titled “Library Management System using C++” as a part of the Computing Aptitude course

I am also thankful to my friends and classmates for their continuous cooperation, constructive feedback, and encouragement during this project.Finally, I would like to express my gratitude to my family for their constant support and inspiration, which helped me complete this project successfully.

Abstract:

The “Library Management System” is a console-based C++ application designed to handle daily library operations efficiently. The system enables users to add new books, display available books, issue or return books, and maintain proper records. This project applies key concepts of C++ such as classes, objects, file handling, and control structures. The aim is to automate basic library functions and reduce manual effort.

Introduction:

Libraries play a vital role in education, but managing records manually often leads to inefficiencies and errors. This project provides a computerized solution that simplifies book management.The system allows adding, searching, issuing, and returning books using a simple text-based interface.The motivation for this project is to understand how programming logic and file handling can be used to store and manipulate real-world data effectively.

Program Focus:

1. The project focuses on developing a simple Library Management System to automate book-related operations like adding, issuing, and returning books.

2. It uses C++ programming to apply concepts of object-oriented programming, file handling, and modular design.

3. The main goal is to simplify library record management, reduce manual work, and ensure efficient data storage and retrieval.

Problem Statement / Objectives

Problem Statement:

Manual record-keeping in libraries is time-consuming and error-prone. There is a need for an automated system to manage book details efficiently.

Objectives:

1. To create a C++ program for managing library book records.

2. To implement key features: add, view, issue, and return books.

3. To apply Object-Oriented Programming concepts for modular design.To use

   file handling for persistent storage of book data.

System Design / Approach

Architecture:

Input Module: Takes book and member details.Processing Module: Handles add, search, issue, and return operations.Storage Module: Stores records using file handling.Output Module: Displays stored or searched data.

Class Diagram:

```
class Book {
 int bookID;
 char title[50];
 char author[50];
 bool issued;
 public:
 void getBookData();
 void showBookData();
 int getBookID();
 void issueBook();
 void returnBook();};
```

Implementation
code (main sections):

```
#include <iostream>
#include <fstream>
#include <iomanip>
#include <cstring>
using namespace std;

class Book {

 int bookID;
 char title[50];
 char author[50];
 bool issued;

public:

  void getBookData() {

 cout << "\nEnter Book ID: "; cin >> bookID;

   cin.ignore();

cout << "Enter Book Title: "; cin.getline(title, 50);

 cout << "Enter Author Name: "; cin.getline(author, 50);

  issued = false;

   }
   void showBookData() {

  cout << setw(10) << bookID << setw(25) << title << setw(25)

  << author << setw(15) << (issued ? "Issued" : "Available") << endl;
   }
int getBookID() { return bookID; }
void issueBook() { issued = true; }
void returnBook() { issued = false; }
 bool isIssued() { return issued; }
};

void addBook() {
Book b;
ofstream outFile("library.dat", ios::binary |
ios::app);
 b.getBookData();
outFile.write((char*)&b, sizeof(b)
);
outFile.close();
cout << "\nBook added successfully!\n";
}

void displayBooks() {
 Book b;
ifstream inFile("library.dat", ios::binary);
cout << "\n--- Library Books ---\n";
cout << setw(10) << "ID" << setw(25) << "Title" << setw(25)      <<
"Author" << setw(15) << "Status\n";
 cout << "------------------------------------------------------------\n";

while (inFile.read((char*)&b, sizeof(b))) {
b.showBookData();
}

inFile.close();

}
```

```
void issueBook() {
int id, found = 0;
Book b;
fstream file("library.dat", ios::binary | ios::in |

ios::out);

cout << "\nEnter Book ID to Issue: ";

cin >> id;

while (file.read((char*)&b, sizeof(b)) && !found) {

if (b.getBookID () == id) {

 found = 1;

 if (!b.isIssued()) {
  b.issueBook();

int pos = -1* sizeof(b);

 file.seekp(pos, ios::cur);

file.write((char*)&b, sizeof(b));

cout << "\nBook issued successfully!\n";

 } else {

cout << "\nBook already issued.\n";

                              }
               }
      }

if (!found) cout << "\nBook not found!\n";
file.close();

}
```

```
void returnBook() {
 int id, found = 0;
Book b;

fstream file("library.dat", ios::binary | ios::in |
ios::out);
cout << "\nEnter Book ID to Return: ";
cin >> id;
while (file.read((char*)&b, sizeof(b)) && !found) {
 if (b.getBookID() == id)
{
  found = 1;
  if (b.isIssued()) {
  b.returnBook();
     int pos = -1* sizeof(b);
 file.seekp(pos, ios::cur);
 file.write((char*)&b, sizeof(b));
 cout << "\nBook returned successfully!\n";
} else {
cout << "\nBook was not issued.\n";
                      }
             }
   }
if (!found) cout << "\nBook not found!\n";
 file.close();
}
int main() {
int choice;
do {      cout << "\n===== LIBRARY MANAGEMENT SYSTEM =====";
cout << "\n1. Add Book\n2. Display Books\n3. Issue Book\n4. Return Book\n5.
Exit";

cout << "\nEnter your choice: ";
 cin >> choice;
switch (choice) {
case 1: addBook(); break;
case 2: displayBooks(); break;
 case 3: issueBook(); break;
case 4: returnBook(); break;
case 5: cout << "\nExiting..."; break;
default: cout << "\nInvalid choice!";
       }
    } while (choice != 5);
 return 0;
}
```

Results / Testing
Sample Output:

```
===== LIBRARY MANAGEMENT SYSTEM =====

1. Add Book

2. Display Books

3. Issue Book

4. Return Book

5. ExitEnter your choice: 1

Enter Book ID: 101

Enter Book Title: C++ Programming

Enter Author Name: Bjarne Stroustrup

Book added successfully
```

Conclusion

This mini project successfully implements a functional Library Management System using C++. It applies fundamental programming concepts such as object-oriented design, file handling, and modular development. The project improves logical reasoning, problem-solving, and code structuring ability

Learning Outcomes:-

By completing this project, the student will be able to:Apply C++ OOP principles (classes, objects, encapsulation).Use file handling for persistent data storage.Implement menu-driven programs with modular design.Analyze and solve real-world problems using computational thinking.Demonstrate software development skills for small-scale systems.