```
1 #include <iostream.h>
2 #include <fstream.h>
3 #include <iomanip.h>
4 #include <stdlib.h>
5 #include <stdio.h>
6
7
8 class Book {
 9     long bookNumber;
10     char bookName[100];
11     char author[50];
12 public:
13     void add_new_book() {
14         cout << "\n\n\t\t\t** ADD NEW BOOK **\n\n";
15         cout << "\nBOOK NUMBER: ";
16         cin >> bookNumber;
17         fflush(stdin);
18         cout << "\nNAME OF THE BOOK: ";
19         gets(bookName);
20         fflush(stdin);
21         cout << "\nNAME OF THE AUTHOR: ";
22         gets(author);
23         fflush(stdin);
24         cout << "\n** BOOK SUCCESSFULLY ADDED! **\n";
25     }
26
27     void view_book() {
28         cout << "\nBOOK NUMBER: " << bookNumber;
29         cout << "\n\nNAME OF THE BOOK: " << bookName;
30         cout << "\n\nNAME OF THE AUTHOR: " << author;
31     }
32
33     void update_book() {
34         cout << "\n\n ** ENTER NEW DETAILS!**\n\n";
35         cout << "\nBOOK NAME: ";
36         gets(bookName);
37         cout << "\nNAME OF THE AUTHOR: ";
38         gets(author);
39     }
40
41     long get_book_number() {
42         return bookNumber;
43     }
44
45     void report() {
```

```
46          cout << endl;
47          cout << setw(10);
48          cout << bookNumber;
49          cout << setw(40);
50          cout << bookName;
51          cout << setw(30);
52          cout << author;
53      }
54  };
55
56
57  class Student {
58      long admissionNumber;
59      char studentName[50];
60      long issuedBookNumber;
61      int booksIssued;
62  public:
63      void add_new_student() {
64          cout << "\n\n\t\t\t** ADD NEW STUDENT **\n\n";
65          cout << "\nADMISSION NUMBER: ";
66          cin >> admissionNumber;
67          fflush(stdin);
68          cout << "\nNAME OF THE STUDENT: ";
69          gets(studentName);
70          fflush(stdin);
71          booksIssued = 0;
72          cout << "\n\n** STUDENT SUCCESSFULLY ADDED! **\n\n
    ";
73      }
74
75      void view_student() {
76          cout << "\nADMISSION NUMBER: " << admissionNumber;
77          cout << "\nNAME OF THE STUDENT: " << studentName;
78          cout << "\nNUMBER OF BOOKS ISSUED: " <<
    booksIssued;
79          if (!booksIssued) {
80              cout << "\n\n** THE STUDENT HAS ISSUED NO
    BOOKS!**\n\n";
81          }
82      }
83
84      void update_student() {
85          cout << "\n\n ** ENTER NEW DETAILS!**\n\n";
86          cout << "\nNAME OF THE STUDENT: " << studentName;
87      }
```

```
 88
 89     long get_admission_number() {
 90         return admissionNumber;
 91     }
 92
 93     long get_issued_books() {
 94         return issuedBookNumber;
 95     }
 96
 97     int get_issued_number() {
 98         return booksIssued;
 99     }
100
101     void add_issued_books() {
102         booksIssued++;
103     }
104
105     void return_book() {
106         booksIssued--;
107     }
108
109     void set_book_number(long book_number) {
110         issuedBookNumber = book_number;
111     }
112
113     void report() {
114         cout << endl;
115         cout << setw(10);
116         cout << admissionNumber;
117         cout << setw(30);
118         cout << studentName;
119         cout << setw(10);
120         cout << booksIssued;
121     }
122 };
123
124
125 fstream f;
126 Book book;
127 Student student;
128
129
130 void new_book() {
131     char choice;
132     f.open("books.dat", ios::app);
```

```
133     do {
134         system("cls");
135         book.add_new_book();
136         f.write((char *) &book, sizeof(Book));
137         cout << "\n\nDO YOU WANT TO ADD MORE RECORDS? [Y/
    n] ";
138         cin >> choice;
139         fflush(stdin);
140     } while (tolower(choice) != 'n');
141     f.close();
142 }
143
144
145 void new_student() {
146     char choice;
147     f.open("students.dat", ios::app);
148     do {
149         system("cls");
150         student.add_new_student();
151         f.write((char *) &student, sizeof(Student));
152         cout << "\n\nDO YOU WANT TO ADD MORE RECORDS? [Y/
    n] ";
153         cin >> choice;
154         fflush(stdin);
155     } while (tolower(choice) != 'n');
156     f.close();
157 }
158
159
160 void display_book(long bookNumber) {
161     system("cls");
162     cout << "\n\n\t\t\t** BOOK DETAILS ** \n\n";
163     int flag = 1;
164     f.open("books.dat", ios::in);
165     while (f.read((char *) &book, sizeof(Book))) {
166         if (bookNumber == book.get_book_number()) {
167             book.view_book();
168             flag = 0;
169         }
170     }
171     f.close();
172     if (flag) {
173         cout << "\n\n** BOOK DOES NOT EXIST! **\n\n";
174     }
175     else {
```

```
176         cout << "\n\n";
177     }
178     system("pause");
179 }
180 
181 
182 void display_student(long admissionNumber) {
183     system("cls");
184     cout << "\n\n\t\t\t** STUDENT DETAILS ** \n";
185     int flag = 1;
186     f.open("students.dat", ios::in);
187     while (f.read((char *) &student, sizeof(Student))) {
188         if (admissionNumber == student.
    get_admission_number()) {
189             student.view_student();
190             flag = 0;
191         }
192     }
193     f.close();
194     if (flag) {
195         cout << "\n\n** STUDENT DOES NOT EXIST! **\n\n";
196     }
197     else {
198         cout << "\n\n";
199     }
200     system("pause");
201 }
202 
203 
204 void update_book_record() {
205     long bookNumber;
206     int found = 0;
207     cout << "\n\n\t\t\t** MODIFY BOOK RECORD **\n\n";
208     cout << "\nBOOK NUMBER OF THE BOOK TO BE UPDATED: ";
209     cin >> bookNumber;
210     fflush(stdin);
211     f.open("books.dat", ios::in | ios::out);
212     while (f.read((char *) &book, sizeof(Book))) {
213         if (book.get_book_number() == bookNumber) {
214             book.view_book();
215             book.update_book();
216             int pos = -1 * sizeof(Book);
217             f.seekp(pos, ios::cur);
218             f.write((char *) &book, sizeof(Book));
219             cout << "\n\n** BOOK SUCCESSFULLY UPDATED! **
```

```
219 \n\n";
220             found = 1;
221         }
222     }
223     f.close();
224     if (!found) {
225         cout << "\n\n** BOOK NOT FOUND!! **\n\n";
226     }
227     system("pause");
228 }
229
230
231 void update_student_record() {
232     long admissionNumber;
233     int found = 0;
234     cout << "\n\n\t\t\t** MODIFY STUDENT RECORD **\n\n";
235     cout << "\nADMISSION NUMBER OF THE STUDENT TO BE
    UPDATED: ";
236     cin >> admissionNumber;
237     fflush(stdin);
238     f.open("student.dat", ios::in | ios::out);
239     while (f.read((char *) &student, sizeof(Student))) {
240         if (student.get_admission_number() ==
    admissionNumber) {
241             student.view_student();
242             student.update_student();
243             int pos = -1 * sizeof(Student);
244             f.seekp(pos, ios::cur);
245             f.write((char *) &student, sizeof(Student));
246             cout << "\n\n** STUDENT SUCCESSFULLY UPDATED
    ! **\n\n";
247             found = 1;
248         }
249     }
250     f.close();
251     if (!found) {
252         cout << "\n\n** STUDENT NOT FOUND!! **\n\n";
253     }
254     system("pause");
255 }
256
257
258 void remove_student() {
259     long admissionNumber;
260     int flag = 0;
```

```
261     cout << "\n\n\t\t\t** DELETE STUDENT **\n\n";
262     cout << "\nADMISSION NUMBER OF THE STUDENT TO BE
    DELETED: ";
263     cin >> admissionNumber;
264     fflush(stdin);
265     f.open("students.dat", ios::in | ios::out);
266     fstream fil;
267     fil.open("temp.dat", ios::out);
268     f.seekg(0, ios::beg);
269     while (f.read((char *) &student, sizeof(Student))) {
270         if (student.get_admission_number() !=
    admissionNumber) {
271             fil.write((char *) &student, sizeof(Student))
    ;
272         }
273         else {
274             flag = 1;
275         }
276     }
277     fil.close();
278     f.close();
279     remove("students.dat");
280     rename("temp.dat", "students.dat");
281     if (flag) {
282         cout << "\n\n** STUDENT-RECORD DELETED!! **\n\n";
283     }
284     else {
285         cout << "\n\n** STUDENT NOT FOUND!! **\n\n";
286     }
287     system("pause");
288 }
289
290
291 void remove_book() {
292     long bookNumber;
293     int flag = 0;
294     cout << "\n\n\t\t\t** DELETE BOOK  **\n\n";
295     cout << "\nBOOK NUMBER OF THE BOOK TO BE DELETED: ";
296     cin >> bookNumber;
297     fflush(stdin);
298     f.open("books.dat", ios::in | ios::out);
299     fstream fil;
300     fil.open("temp.dat", ios::out);
301     f.seekg(0, ios::beg);
302     while (f.read((char *) &book, sizeof(Book))) {
```

```
303         if (book.get_book_number() != bookNumber) {
304             fil.write((char *) &book, sizeof(Book));
305         }
306         else {
307             flag = 1;
308         }
309     }
310     fil.close();
311     f.close();
312     remove("books.dat");
313     rename("temp.dat", "books.dat");
314     if (flag) {
315         cout << "\n\n** BOOK-RECORD DELETED!! **\n\n";
316     }
317     else {
318         cout << "\n\n** BOOK NOT FOUND!! **\n\n";
319     }
320     system("pause");
321 }
322
323
324 void student_list() {
325     f.open("students.dat", ios::in);
326     if (!f) {
327         cout << "\n\n** THERE ARE NO STUDENTS IN THE
    SYSTEM! **\n\n";
328         cout << "\n\n";
329         system("pause");
330         return;
331     }
332
333     cout << "\n\n\t ** LIST OF THE STUDENTS IN THE SYSTEM
     **\n\n";
334     cout <<
    "==========================================================
    ==\n";
335     cout << "ADMISSION NUMBER" << "     " << "NAME OF THE
     STUDENT" << "     " << "BOOKS ISSUED\n";
336     cout <<
    "==========================================================
    ==\n";
337
338     while (f.read((char *) &student, sizeof(Student))) {
339         student.report();
340     }
```

```
341     f.close();
342     cout << "\n\n";
343     system("pause");
344 }
345 
346 
347 void book_list() {
348     f.open("books.dat", ios::in);
349     if (!f) {
350         cout << "\n\n** THERE ARE NO BOOKS IN THE SYSTEM
    ! **\n\n";
351         cout << "\n\n";
352         system("pause");
353         return;
354     }
355     cout << "\n\n\t\t\t** LIST OF THE BOOKS IN THE SYSTEM
     **\n\n";
356     cout <<
    "================================================================
    ================================\n";
357     cout << "-- BOOK NUMBER --" << "              " <<
    "-- NAME OF THE BOOK --" << "            " << "-- NAME OF
    THE AUTHOR --\n";
358     cout <<
    "================================================================
    ================================\n";
359 
360     while (f.read((char *) &book, sizeof(Book))) {
361         book.report();
362     }
363     f.close();
364     cout << "\n\n";
365     system("pause");
366 }
367 
368 
369 void issue_book() {
370     long admissionNumber, bookNumber;
371     int found = 0, flag = 0;
372     fstream fil;
373     cout << "\n\n\t\t** BOOK ISSUE **\n\n";
374     cout << "\n\nADMISSION NUMBER OF THE STUDENT: ";
375     cin >> admissionNumber;
376     fflush(stdin);
377     f.open("students.dat", ios::in | ios::out);
```

```
    fil.open("books.dat", ios::in | ios::out);
    while (f.read((char *) &student, sizeof(Student))) {
        if (student.get_admission_number() ==
admissionNumber) {
            found = 1;
            if (student.get_issued_number() < 1) {
                cout << "\n\nBOOK NUMBER OF THE BOOK TO
BE ISSUED: ";
                cin >> bookNumber;
                while (fil.read((char *) &book, sizeof(
Book))) {
                    if (book.get_book_number() ==
bookNumber) {
                        book.view_book();
                        flag = 1;
                        student.add_issued_books();
                        student.set_book_number(
bookNumber);
                        int pos = -1 * sizeof(Student);
                        f.seekp(pos, ios::cur);
                        f.write((char *) &student, sizeof
(Student));
                        cout << "\n\n** BOOK ISSUED
SUCCESSFULLY! **\n\n";
                    }
                }
                if (!flag) {
                    cout << "\n\n** BOOK NUMBER DOES NOT
EXIST! **\n\n";
                }
            }
            else {
                cout << "\n\n** YOU CAN ONLY ISSUE ONE
BOOK AT A TIME! **\n\n";
            }
        }
    }
    if (!found) {
        cout << "\n\n** STUDENT DOESN'T EXIST! **\n\n";
    }
    system("pause");
    f.close();
    fil.close();
}

```

```
414
415 void return_book() {
416     long bookNumber, admissionNumber;
417     int found = 0, flag = 0;
418     fstream fil;
419     system("cls");
420     cout << "\n\n\t\t\t** BOOK DEPOSIT **\n\n";
421     cout << "\n\nENTER ADMISSION NUMBER: ";
422     cin >> admissionNumber;
423     fflush(stdin);
424     f.open("students.dat", ios::in | ios::out);
425     fil.open("books.dat", ios::in | ios::out);
426     while (f.read((char *) &student, sizeof(Student))) {
427         if (student.get_admission_number() ==
    admissionNumber) {
428             found = 1;
429             if (student.get_issued_number() == 1) {
430                 while (fil.read((char *) &book, sizeof(
    Book))) {
431                     if (student.get_issued_books() ==
    book.get_book_number()) {
432                         book.view_book();
433                         flag = 1;
434                         student.return_book();
435                         int pos = -1 * sizeof(student);
436                         f.seekp(pos, ios::cur);
437                         f.write((char *) &student, sizeof
    (Student));
438                         cout << "\n\n** BOOK DEPOSITED
    SUCCESSFULLY! **\n\n";
439                     }
440                 }
441                 if (!flag) {
442                     cout << "\n\n** BOOK NUMBER DOES NOT
    EXIST! **\n\n";
443                 }
444             }
445             else {
446                 cout << "\n\n** NO BOOKS ISSUED. PLEASE
    CHECK! **\n\n";
447             }
448
449         }
450     }
451     if (!found) {
```

```
452         cout << "\n\n** STUDENT DOESN'T EXIST! **\n\n";
453     }
454     system("pause");
455     f.close();
456     fil.close();
457 }
458 
459 
460 void administrator() {
461     int choice;
462     cout << "\n\n\n\t\t** ADMINISTRATOR MENU **";
463     cout << "\n\n\t1- ADD STUDENT";
464     cout << "\n\n\t2- DISPLAY ALL STUDENTS";
465     cout << "\n\n\t3- DISPLAY SPECIFIC STUDENT";
466     cout << "\n\n\t4- UPDATE STUDENT RECORD";
467     cout << "\n\n\t5- DELETE STUDENT";
468     cout << "\n\n\t6- ADD BOOK";
469     cout << "\n\n\t7- DISPLAY ALL BOOKS";
470     cout << "\n\n\t8- DISPLAY SPECIFIC BOOK";
471     cout << "\n\n\t9- UPDATE A BOOK";
472     cout << "\n\n\t10- DELETE BOOK";
473     cout << "\n\n\t11- BACK TO MAIN MENU";
474     cout << "\n\n\tENTER YOUR CHOICE: ";
475     cin >> choice;
476     switch (choice) {
477         case 1:
478             system("cls");
479             new_student();
480             break;
481         case 2:
482             system("cls");
483             student_list();
484             break;
485         case 3:
486             long admissionNumber;
487             system("cls");
488             cout << "\n\n\t\t** STUDENT-DETAILS ** \n\n";
489             cout << "\n\nADMISSION NUMBER: ";
490             cin >> admissionNumber;
491             fflush(stdin);
492             display_student(admissionNumber);
493             break;
494         case 4:
495             system("cls");
496             update_student_record();
```

```
497             break;
498         case 5:
499             system("cls");
500             remove_student();
501             break;
502         case 6:
503             system("cls");
504             new_book();
505             break;
506         case 7:
507             system("cls");
508             book_list();
509             break;
510         case 8:
511             long bookNumber;
512             system("cls");
513             cout << "\n\n\t\t** BOOK-DETAILS ** \n\n";
514             cout << "\n\nBOOK NUMBER: ";
515             cin >> bookNumber;
516             fflush(stdin);
517             display_book(bookNumber);
518             break;
519         case 9:
520             system("cls");
521             update_book_record();
522             break;
523         case 10:
524             remove_book();
525             break;
526         case 11:
527             return;
528         default:
529             cout << "\a";
530     }
531     system("cls");
532     administrator();
533 }
534
535
536 int main() {
537     char ch;
538     do {
539         system("cls");
540         cout << "\n\n\t\t\t** LIBRARY MANAGEMENT SYSTEM
    **\n\n";
```

```
541         cout << "\n\n** MAIN MENU **";
542         cout << "\n\n\t1- ISSUE BOOK";
543         cout << "\n\n\t2- RETURN A BOOK";
544         cout << "\n\n\t3- ADMINISTRATOR\'S MENU";
545         cout << "\n\n\t4- EXIT";
546         cout << "\n\n\tCHOICE: ";
547         cin >> ch;
548         fflush(stdin);
549         switch (ch) {
550             case '1':
551                 system("cls");
552                 issue_book();
553                 break;
554             case '2':
555                 system("cls");
556                 return_book();
557                 break;
558             case '3':
559                 system("cls");
560                 administrator();
561                 break;
562             case '4':
563                 exit(0);
564             default :
565                 cout << "\a";
566         }
567     } while (ch != '4');
568     return 0;
569 }
```