

DATA STRUCTURES LABORATORY

Course Code: 1BCSL306

VTU 2025 Scheme – Semester 3

EXPERIMENT – 01

Library Book Management

Structures and Dynamic Memory Allocation

**Simple Code • Easy to Remember • Easy to Write
in Exam**

SEARCH CREATORS

Your VTU Study Partner

Experiment 01

1. Aim

Aim

To develop a menu-driven C program to manage library book records using structures and dynamic memory allocation.

SEARCH CREATORS

2. VTU Prescribed Experiment

VTU Prescribed Experiment

Develop a Program in C for the following:

- a. Define a record (structure) titled **Book** with the following fields:
 - Book_ID
 - Title
 - Author
 - Price
 - Availability Status (Available / Issued)
- b. Dynamically allocate memory to store the details of N books, where N is entered by the user.
- c. Implement the following functions:
 - i. **create()** – To allocate memory and read the details of N books and store them in the allocated memory.
 - ii. **display()** – To display the details of all books available in the library.
 - iii. **search()** – To search for a book using Book ID and display the book details if found.
 - iv. **issueBook()** – To issue a book by updating its availability status and display an appropriate message.
 - v. **returnBook()** – To return an issued book by updating its availability status and display an appropriate message.
- d. Build a menu-driven interface with the following options:
 - i. Add Book Records
 - ii. Display all Book Records
 - iii. Search Book by Book ID
 - iv. Issue a Book
 - v. Return a Book
 - vi. Exit

3. Simple Theory

Structure:

A structure is a user-defined data type used to store related data of different types under one name.

Here, one book record contains:

Book ID + Title + Author + Price + Status

Dynamic Memory Allocation:

Memory is allocated during program execution using:

malloc()

Memory is released using:

free()

In this program:

1 = Available

0 = Issued

4. Algorithm

1. Start the program.
2. Define structure **Book**.
3. Read the number of books N .
4. Allocate memory for N books using `malloc()`.
5. Read:
 - Book ID
 - Title
 - Author
 - Price
6. Set book status as Available.
7. Display the menu.
8. According to the user's choice:
 - Create book records.
 - Display all books.
 - Search a book.
 - Issue a book.

- Return a book.
9. Repeat until Exit is selected.
 10. Free the allocated memory.
 11. Stop.

Easy Exam Memory

Remember only these five functions:

$create() \rightarrow display() \rightarrow search() \rightarrow issueBook() \rightarrow returnBook()$

For issue and return:

$1 = Available$

$0 = Issued$

5. C Program – Easy Student Version

```
1 #include<stdio.h>
2 #include<stdlib.h>
3
4 struct Book
5 {
6     int id;
7     char title[30];
8     char author[30];
9     float price;
10    int status;
11 };
12
13 struct Book *b;
14 int n;
15
16 void create()
17 {
18     int i;
19
20     printf("Enter number of books: ");
21     scanf("%d",&n);
22
23     b = (struct Book*)malloc(n*sizeof(struct Book));
24
25     for(i=0;i<n;i++)
26     {
27         printf("\nEnter Book %d details\n",i+1);
28
29         printf("Book ID: ");
30         scanf("%d",&b[i].id);
31
32         printf("Title: ");
33         scanf(" %[^\n]",b[i].title);
34
35         printf("Author: ");
36         scanf(" %[^\n]",b[i].author);
37
38         printf("Price: ");
39         scanf("%f",&b[i].price);
40
41         b[i].status = 1;
42     }
43 }
44
45 void display()
46 {
47     int i;
48
49     for(i=0;i<n;i++)
50     {
```

```
51     printf("\nBook ID: %d",b[i].id);
52     printf("\nTitle: %s",b[i].title);
53     printf("\nAuthor: %s",b[i].author);
54     printf("\nPrice: %.2f",b[i].price);
55
56     if(b[i].status==1)
57         printf("\nStatus: Available\n");
58     else
59         printf("\nStatus: Issued\n");
60 }
61 }
62
63 void search()
64 {
65     int id,i;
66
67     printf("Enter Book ID: ");
68     scanf("%d",&id);
69
70     for(i=0;i<n;i++)
71     {
72         if(b[i].id==id)
73         {
74             printf("\nBook Found");
75             printf("\nID: %d",b[i].id);
76             printf("\nTitle: %s",b[i].title);
77             printf("\nAuthor: %s",b[i].author);
78             printf("\nPrice: %.2f\n",b[i].price);
79             return;
80         }
81     }
82
83     printf("Book Not Found\n");
84 }
85
86 void issueBook()
87 {
88     int id,i;
89
90     printf("Enter Book ID to Issue: ");
91     scanf("%d",&id);
92
93     for(i=0;i<n;i++)
94     {
95         if(b[i].id==id)
96         {
97             if(b[i].status==1)
98             {
99                 b[i].status=0;
100                 printf("Book Issued Successfully\n");
101             }
102             else
103                 printf("Book Already Issued\n");
```

```
104         return;
105     }
106 }
107
108
109     printf("Book Not Found\n");
110 }
111
112 void returnBook()
113 {
114     int id,i;
115
116     printf("Enter Book ID to Return: ");
117     scanf("%d",&id);
118
119     for(i=0;i<n;i++)
120     {
121         if(b[i].id==id)
122         {
123             if(b[i].status==0)
124             {
125                 b[i].status=1;
126                 printf("Book Returned Successfully\n");
127             }
128             else
129                 printf("Book Already Available\n");
130
131             return;
132         }
133     }
134
135     printf("Book Not Found\n");
136 }
137
138 int main()
139 {
140     int ch;
141
142     do
143     {
144         printf("\n1. Add Book Records");
145         printf("\n2. Display all Book Records");
146         printf("\n3. Search Book by Book ID");
147         printf("\n4. Issue a Book");
148         printf("\n5. Return a Book");
149         printf("\n6. Exit");
150
151         printf("\nEnter Choice: ");
152         scanf("%d",&ch);
153
154         switch(ch)
155         {
156             case 1:
```

```
157         create();
158         break;
159
160     case 2:
161         display();
162         break;
163
164     case 3:
165         search();
166         break;
167
168     case 4:
169         issueBook();
170         break;
171
172     case 5:
173         returnBook();
174         break;
175
176     case 6:
177         free(b);
178         printf("Program Ended\n");
179         break;
180
181     default:
182         printf("Invalid Choice\n");
183 }
184
185 }while(ch!=6);
186
187 return 0;
188 }
```

6. Simple Program Explanation

6.1. Book Structure

```
1 struct Book
2 {
3     int id;
4     char title[30];
5     char author[30];
6     float price;
7     int status;
8 };
```

This structure stores all details of one book.

6.2. Book Pointer

```
1 struct Book *b;
```

The pointer `b` stores the address of the dynamically allocated book records.

6.3. Dynamic Memory Allocation

```
1 b = (struct Book*)malloc(n*sizeof(struct Book));
```

This allocates memory for N books.

6.4. create() Function

The `create()` function:

- Reads number of books.
- Allocates memory.
- Reads all book details.
- Sets every new book as Available.

6.5. display() Function

The `display()` function displays:

- Book ID
- Title
- Author
- Price
- Availability Status

6.6. search() Function

The function compares the entered Book ID with every stored Book ID.

Therefore:

Linear Search

is used.

6.7. issueBook() Function

If the book is available:

$$status = 1$$

it is changed to:

$$status = 0$$

Therefore:

Available $\rightarrow$ Issued

6.8. returnBook() Function

If the book is issued:

$$status = 0$$

it is changed to:

$$status = 1$$

Therefore:

Issued $\rightarrow$ Available

Easy Exam Memory

For **Search, Issue and Return**, remember the same logic:

Read ID $\rightarrow$ for loop $\rightarrow$ compare ID $\rightarrow$ perform operation

This makes the program much easier to remember during the lab exam.

7. Sample Output

```
1
2 1. Add Book Records
3 2. Display all Book Records
4 3. Search Book by Book ID
5 4. Issue a Book
6 5. Return a Book
7 6. Exit
8
9 Enter Choice: 1
10
11 Enter number of books: 2
12
13 Enter Book 1 details
14 Book ID: 101
15 Title: Data Structures
16 Author: Horowitz
17 Price: 550
18
19 Enter Book 2 details
20 Book ID: 102
21 Title: C Programming
22 Author: Ritchie
23 Price: 450
24
25
26 Enter Choice: 2
27
28 Book ID: 101
29 Title: Data Structures
30 Author: Horowitz
31 Price: 550.00
32 Status: Available
33
34 Book ID: 102
35 Title: C Programming
36 Author: Ritchie
37 Price: 450.00
38 Status: Available
39
40
41 Enter Choice: 3
42
43 Enter Book ID: 101
44
45 Book Found
46 ID: 101
47 Title: Data Structures
48 Author: Horowitz
49 Price: 550.00
50
```

```
51
52 Enter Choice: 4
53
54 Enter Book ID to Issue: 101
55
56 Book Issued Successfully
57
58
59 Enter Choice: 2
60
61 Book ID: 101
62 Title: Data Structures
63 Author: Horowitz
64 Price: 550.00
65 Status: Issued
66
67
68 Enter Choice: 5
69
70 Enter Book ID to Return: 101
71
72 Book Returned Successfully
73
74
75 Enter Choice: 6
76
77 Program Ended
```

8. Simple Test Cases

1. Create two book records and check whether both records are stored.
2. Search for an existing Book ID.
3. Search for a Book ID that is not present.
4. Issue an available book.
5. Try to issue the same book again.
6. Return an issued book.
7. Try to return an already available book.

9. Result

Result

Thus, the C program for managing library book records using **structures and dynamic memory allocation** was successfully implemented.

The program performs:

- Add Book Records
- Display all Book Records
- Search Book by Book ID
- Issue a Book
- Return a Book

10. Important Viva Questions

Important Viva Questions

1. What is a structure?

A structure is a user-defined data type used to store different types of related data.

2. Why is structure used in this program?

It is used to store Book ID, Title, Author, Price and Status together.

3. What is dynamic memory allocation?

It is the process of allocating memory during program execution.

4. Which function is used for dynamic memory allocation?

`malloc()`

5. Which header file is required for malloc()?

`stdlib.h`

6. Why is malloc() used here?

Because the number of books is entered by the user during program execution.

7. What does free() do?

It releases dynamically allocated memory.

8. What does status = 1 represent?

It means the book is Available.

9. What does status = 0 represent?

It means the book is Issued.

10. Which search technique is used?

Linear Search.

11. What happens when a book is issued?

The status changes from 1 to 0.

12. What happens when a book is returned?

The status changes from 0 to 1.

13. What is a pointer?

A pointer is a variable that stores the address of another variable.

14. What is the purpose of the pointer b?

It stores the address of dynamically allocated Book records.

15. Why is the program menu-driven?

It allows the user to select and perform different operations repeatedly.

11. 1-Minute Exam Revision

Easy Exam Memory

Structure Fields

ID + Title + Author + Price + Status

Dynamic Memory

malloc()

Release Memory

free()

Functions

create()

display()

search()

issueBook()

returnBook()

Status

1 = Available

0 = Issued

Searching

Linear Search

12. Easy Code Memory Map

Remember

Remember the program in this order:

Header → Structure → Pointer → Functions → Main

Then remember:

Search, Issue, Return

all use almost the same logic:

Read ID → Loop → Compare → Action

SEARCH CREATORS

1BCSL306 – Data Structures Laboratory

EXPERIMENT 01 COMPLETED

Library Book Management

Structures and Dynamic Memory Allocation

Understand → Remember → Write → Execute →
Viva

Your VTU Study Partner