



CSE1022 – Introduction to Programming - Laboratory

Laboratory # L49 + L50

Student Name: Chandra Sekhar U	Roll Number: 23BCE9222
---------------------------------------	-------------------------------

List of Programs – Covered During Session/To be done by Own

1. Write a C program to understand structure and nested structures.
2. Write a C program to understand structure with pointers and functions.
3. Define a structure STUDENT to store the following data for a student: name (null-terminated string of length at most 20 chars), roll no. (integer), CGPA (float). Then
 - a. In main, declare an array of 100 STUDENT structures. Read an integer n and then read in the details of n students in this array
 - b. Write a function to search the array for a student by name. Returns the structure for the student if found. If not found, return a special structure with the name field set to empty string (just a '\0')
 - c. Write a function to search the array for a student by rollno.
 - d. Write a function to print the details of all students with CGPA > x for a given x
 - e. Call the functions from the main after reading in name/rollno/CGPA to search

For every program in the above list, you need to follow the format as given on the next page. Every program must start from a new page as shown next.

Program # 1

I. Objective

Write a C program to understand structure and nested structures.

II. Programs Code

```
#include <stdio.h>
#include <string.h>

struct Person {
    char name[50];
    char dob[25];
    char address[50];
    long int mob_no;
};

typedef struct Account{
    long int salary;
    float tax;
}acc;

typedef struct Employee {
    int id;
    struct Person p;
    acc a;
}emp;

float cal_tax(long int s) {
    return (s * 0.2 + 500);
}

int main() {
    printf("Size of struct Person: %zu bytes\n", sizeof(struct Employee));
    struct Person p1 = {"Mr. XYZ", "01-01-2000", "CB-G19, VIT-AP University",
1234567890L};
    acc a1 = {75000, 0.0};
    emp e1;

    e1.id = 4040;
    e1.p = p1;
    e1.a = a1;
    e1.a.tax = cal_tax(e1.a.salary);

    printf("\nID : %d", e1.id);
    printf("\nName : %s", e1.p.name);
    printf("\nDOB : %s", e1.p.dob);
```

```
printf("\nAddress : %s", e1.p.address);  
printf("\nMobile : %ld", e1.p.mob_no);  
printf("\nSalary : %ld", e1.a.salary);  
printf("\nTax : %f", e1.a.tax);
```

```
return 0;
```

```
}
```


III. Results

- Structures to store an employee's personal and financial details, calculates tax based on the employee's salary using a custom formula, and prints the employee's information.

IV. Outcome and Understanding

- From this code, I understand how to use nested structures in c programming

V. Attachments (Screenshots)



```
Size of struct Person: 144 bytes  
ID : 4040  
Name : Mr. XYZ  
DOB : 01-01-2000  
Address : CB-619, VIT-AP University  
Mobile : 1234567890  
Salary : 75000  
Tax : 15500.000000
```

Program # 2

I. Objective

Write a C program to understand structure with pointers and functions.

II. Programs Code

```
#include <stdio.h>
```

```
#define MAX_B 3
```

```
#define MAX_U 2
```

```
typedef struct { char t[30], a[20]; int bid, co, uid; } B;
```

```
typedef struct { char n[30]; int id; } U;
```

```
U *valid_user(int id, U* us) { // return type -- pointer of structure data type
```

```
    int i;
```

```
    for (i = 0; i < MAX_U; i++)
```

```
    if (us[i].id == id) return &us[i];
```

```
    return NULL;
```

```
}
```

```
B *valid_book(int bid, B* lib) {
```

```
    int i;
```

```
    for (i = 0; i < MAX_B; i++)
```

```
    if (lib[i].bid == bid) return &lib[i];
```

```
    return NULL;
```

```
}
```

```
void chk_out(B* b, U* u) {
```

```
    if (!b->co) {
```

```
    b->co = 1; b->uid = u->id;
```

```
    printf("%s checked out \"%s\"\n", u->n, b->t);
```

```
    } else {
```

```
    printf("\"%s\" is already checked out.\n", b->t);
```

```
    }
```

```
}
```

```
void rtn(B* b, U* u) {
```

```
    if (b->co && b->uid == u->id) {
```

```
    b->co = 0; b->uid = 0;
```

```
    printf("%s returned \"%s\"\n", u->n, b->t);
```

```
    } else {
```

```
    printf("Invalid return request.\n");
```

```
    }
```

```
}
```

```
int show_avail(B* lib) {
```

```

    int i,count=0;
    printf("\nAvailable books:\n");
    for (i = 0; i < MAX_B; i++)
if (!lib[i].co){
    printf("%d: \"%s\" by %s\n", lib[i].bid, lib[i].t, lib[i].a);
    count +=1;
}
    return count;
}

int show_issued(B* lib, U* u) {
    int i,count=0;
    printf("\nIssued books:\n");
    for (i = 0; i < MAX_B; i++)
if (lib[i].co && lib[i].uid == u->id){
    printf("%d: \"%s\" by %s\n", lib[i].bid, lib[i].t, lib[i].a);
    count +=1;
}
    return count;
}

int main() {
    B lib[MAX_B] = {
{"1984", "Orwell", 100, 0, 0},
{"Mockingbird", "Lee", 150, 0, 0},
{"Gatsby", "Fitz", 165, 0, 0}
    };

    U us[MAX_U] = {"Alice", 1001}, {"Bob", 1002}};
    int act, uid, bid, count;
    U *u;
    B *b;
    while (1) {
printf("\n1. Check out\n2. Return\n3. Exit\nChoose an action: ");
scanf("%d", &act);

if (act == 3) break;

printf("User ID: ");
scanf("%d", &uid);
u = valid_user(uid, us);
if (!u) {
    printf("Invalid User ID!\n");
    getchar();
    continue;
}

switch (act) {
    case 1: count = show_avail(lib);
        if(count){
            printf("Choose book id to check out: ");
            scanf("%d", &bid);
            b = valid_book(bid, lib);

```

```

        if (b)
            chk_out(b, u);
        else
            printf("Invalid book selection.\n");
    }
    else
        printf("No book available.\n");
    getchar();
    break;
case 2: count = show_issued(lib, u);
    if(count){
        printf("Choose book id to return: ");
        scanf("%d", &bid);
        b = valid_book(bid, lib);
        if (b)
            rtn(b, u);
        else
            printf("Invalid book selection.\n");
    }
    else
        printf("No book issued.\n");
    getchar();
    break;
default: printf("Invalid action!\n");
    getchar();
}
}
return 0;
}

```

III. Results

- A library system where users can check out and return books. It verifies user and book IDs, displays available and issued books, and updates the status of books.

IV. Outcome and Understanding

- From this code, I understand how to use structure with pointers in c programming.

V. Attachments (Screenshots)

```
1. Check out
2. Return
3. Exit
Choose an action: 1
User ID: 1001

Available books:
100: "1984" by Orwell
150: "Mockingbird" by Lee
165: "Gatsby" by Fitz
Choose book id to check out: 100
Alice checked out "1984"

1. Check out
2. Return
3. Exit
Choose an action: 1
User ID: 150
Invalid User ID!

1. Check out
2. Return
3. Exit
Choose an action: 1
User ID: 1001

Available books:
150: "Mockingbird" by Lee
165: "Gatsby" by Fitz
Choose book id to check out: 165
Alice checked out "Gatsby"

1. Check out
2. Return
3. Exit
Choose an action: 3
```

Program # 3

I. Objective

Define a structure STUDENT to store the following data for a student: name (null-terminated string of length at most 20 chars), roll no. (integer), CGPA (float). Then

- a. In main, declare an array of 100 STUDENT structures. Read an integer n and then read in the details of n students in this array
- a. Write a function to search the array for a student by name. Returns the structure for the student if found. If not found, return a special structure with the name field set to empty string (just a '\0')
- a. Write a function to search the array for a student by rollno.
- a. Write a function to print the details of all students with CGPA > x for a given x
- a. Call the functions from the main after reading in name/rollno/CGPA to search

II. Programs Code

```
#include <stdio.h>
#include <string.h>

#define MAX_ENTRIES 100

struct Student {
    char full_name[21];
    int roll_number;
    float grade_point_average;
};

struct Student get_student_by_name(struct Student list[], int count, char* name) {
    for (int i = 0; i < count; i++) {
        if (strcmp(list[i].full_name, name) == 0) {
            return list[i];
        }
    }
    struct Student not_found = {"\0",-1,0.0};
    return not_found;
}

struct Student get_student_by_roll(struct Student list[], int count, int roll) {
    for (int i = 0; i < count; i++) {
        if (list[i].roll_number == roll) {
            return list[i];
        }
    }
    struct Student not_found = {"\0",-1,0.0};
    return not_found;
}
```



```

void print_students_with_high_cgpa(struct Student list[], int count, float min_cgpa)
{
    for (int i = 0; i < count; i++) {
        if (list[i].grade_point_average > min_cgpa) {
            printf("Name: %s, Roll No: %d, CGPA: %.2f\n", list[i].full_name,
list[i].roll_number, list[i].grade_point_average);
        }
    }
}

```

```

int main() {
    struct Student records[MAX_ENTRIES];
    int num_students;

    printf("Enter the number of students: ");
    scanf("%d", &num_students);

    for (int i = 0; i < num_students; i++) {
        printf("Enter full name, roll number, and CGPA for student %d:\n", i + 1);
        scanf("%s %d %f", records[i].full_name, &records[i].roll_number,
&records[i].grade_point_average);
    }
}

```

```

int choice;
do {
    printf("\nMenu:\n");
    printf("1. Search by Name\n");
    printf("2. Search by Roll Number\n");
    printf("3. Print Students with CGPA Greater Than X\n");
    printf("4. Exit\n");
    printf("Enter your choice: ");
    scanf("%d", &choice);

    switch (choice) {
        case 1: {
            char search_full_name[21];
            printf("Enter the name to search: ");
            scanf("%s", search_full_name);
            struct Student found_by_name = get_student_by_name(records,
num_students, search_full_name);
            if (found_by_name.full_name[0] == '\0') {
                printf("No student found with the name '%s'.\n", search_full_name);
            } else {
                printf("Student found: Name: %s, Roll No: %d, CGPA: %.2f\n",
found_by_name.full_name, found_by_name.roll_number,
found_by_name.grade_point_average);
            }
            break;
        }
        case 2: {
            int search_roll_number;
            printf("Enter the roll number to search: ");
            scanf("%d", &search_roll_number);

```

```

        struct Student found_by_roll = get_student_by_roll(records,
num_students, search_roll_number);
        if (found_by_roll.full_name[0] == '\0') {
            printf("No student found with the roll number %d.\n",
search_roll_number);
        } else {
            printf("Student found: Name: %s, Roll No: %d, CGPA: %.2f\n",
found_by_roll.full_name, found_by_roll.roll_number,
found_by_roll.grade_point_average);
        }
        break;
    }
    case 3: {
        float cgpa_threshold;
        printf("Enter the CGPA threshold: ");
        scanf("%f", &cgpa_threshold);
        printf("Students with CGPA greater than %.2f:\n", cgpa_threshold);
        print_students_with_high_cgpa(records, num_students, cgpa_threshold);
        break;
    }
    case 4:
        printf("Exiting program.\n");
        break;
    default:
        printf("Invalid choice! Please try again.\n");
    }
} while (choice != 4);

return 0;
}

```

III. Results

This is program for student records, allowing users to input details like name, roll number, and CGPA for a specified number of students. It provides a menu for searching students by name or roll number.

IV. Outcome and Understanding

- From this code, I understand how use all basic concepts(structures, switch, nested structures) in c programming language.

V. Attachments (Screenshots)

```
Enter the number of students: 2
Enter full name, roll number, and CGPA for student 1:
ram
101
9
Enter full name, roll number, and CGPA for student 2:
xjay
102
10
Menu:
1. Search by Name
2. Search by Roll Number
3. Print Students with CGPA Greater Than X
4. Exit
Enter your choice: 1
Enter the name to search: ram
Student found: Name: ram, Roll No: 101, CGPA: 9.00
Menu:
1. Search by Name
2. Search by Roll Number
3. Print Students with CGPA Greater Than X
4. Exit
Enter your choice: 2
Enter the roll number to search: 101
Student found: Name: ram, Roll No: 101, CGPA: 9.00
Menu:
1. Search by Name
2. Search by Roll Number
3. Print Students with CGPA Greater Than X
4. Exit
Enter your choice: 3
Enter the CGPA threshold: 9
Students with CGPA greater than 9.00:
Name: xjay, Roll No: 102, CGPA: 10.00
Menu:
1. Search by Name
2. Search by Roll Number
3. Print Students with CGPA Greater Than X
4. Exit
Enter your choice: 4
Exiting program.
```