
LAB MANUAL

for

PROGRAMMING IN C

Paper Code: (CC -2)

1st Semester

B.Sc. in Computer Science (Hons.)



Prepared by

RANU CHOWDHURY
Department of Comp. Sc.
SIR GURUDAS MAHAVIDYALAYA

OBJECTIVES

1. To introduce students to the basic knowledge of programming fundamentals of C language.
2. To impart writing skill of C programming to the students and solving problems.
3. To impart the concepts like looping, array, functions, pointers, file, structure.

COURSE OUTCOME

After completing this lab course you will be able to:

1. Understand the logic for a given problem.
2. Write the algorithm of a given problem.
3. Draw a flow chart of a given problem.
4. Recognize and understand the syntax and construction of C programming code.
5. Gain experience of procedural language programming.
6. Know the steps involved in compiling, linking and debugging C code.
7. Understand using header files.
8. Learn the methods of iteration or looping and branching.
9. Make use of different data-structures like arrays, pointers, structures and files.
10. Understand how to access and use library functions.
11. Understand function declaration and definition.
12. Understand proper use of user defined functions.
13. Write programs to print output on the screen as well as in the files.
14. Apply all the concepts that have been covered in the theory course, and
15. Know the alternative ways of providing solution to a given problem.

INTRODUCTION ABOUT LAB

Steps involved in program development:-

To develop the program in high level language and translate it into machine level language following steps have to be practiced.

1. Writing and editing the program.
 2. Linking the program with the required library modules.
 3. Compiling the program.
 4. Executing the program.
-

Algorithm:-

It is a method of representing the step by step process for solving a problem. Each step is called an instruction.

Characteristics of algorithm are:

- **Finiteness:-** It terminates with finite number of steps.
- **Definiteness:-** Each step of algorithm is exactly defined.
- **Effectiveness:-** All the operations used in the algorithm can be performed exactly in a fixed duration of time.
- **Input:-** An algorithm must have an input before the execution of program begins.
- **Output:-** An algorithm has one or more outputs after the execution of the program.

Example of algorithm to find sum of two numbers:

Step1: BEGIN

Step2: READ a, b

Step3: ADD a and b and store in variable c

Step4: DISPLAY c

Step5: STOP

ABOUT C LANGUAGE

C is a programming language developed by Dennis Ritchie at AT&T's BELL Laboratory of USA in 1972. Because of its reliability, C is very popular. C is highly portable & it is well suited for structured programming. C program consists of collection of functions.

HISTORY OF C

The milestones in C's development as a language are listed below:

1. BCPL - a user friendly OS providing powerful development tools developed from BCPL c. 1967. Assembler was tedious, long and error prone.
2. A new language "B" was attempted c. 1970 by Ken Thompson at Bell Labs.
3. UNIX was developed c. 1970 using "B" - DEC PDP-7 Assembly Language.
4. A totally new language "C", a successor to "B" was developed c. 1971.
5. By 1973, UNIX OS was almost totally written in "C".

Hardware Requirement: Desktop Computer / laptop computer.

Software Requirement: Linux Operating System with GCC / TURBO C in WINDOWS OS / TURBO C++ in WINDOWS OS.

GCC

GCC is a Linux-based C compiler released by the Free Software Foundation which is usually operated via the command line. It often comes distributed freely with a Linux installation, so if you are running UNIX or a Linux variant you will probably have it on your system. You can invoke GCC on a source code file simply by typing:-

```
gcc filename
```

The default executable output of GCC is "a.out", which can be run by typing **"/a.out"**. It is also possible to specify a name for the executable file at the command line by using the syntax **"-o outputfile"**, as shown in the following example :-

```
gcc filename -o outputfile
```

Again, you can run your program with **"/outputfile"**. (The **/** is there to ensure you run the program for the current working directory.)

Note: If you need to use functions from the math library (generally functions from "math.h" such as sin or sqrt), then you need to explicitly ask it to link with that library with the **"-l"** flag and the library **"m"**:

```
gcc filename -o outputfile -lm
```

Turbo C/C++

Open Turbo C/C++ from your Desktop or Programs menu. Select “**File**” from Menu bar and select option “**New**” and Save C program with filename „.C” extension.

To do compiling – **Select -> Compile** from menu and **click-> compile**.

If the compilation is successful – you will see a “**success**” message. Else you will see the number of errors.

To RUN the program – you may **select ->Run** from menu and **click -> Run**

Now you will see the output screen.

STRUCTURE OF „C” PROGRAM :

C program is a collection of several instructions where each instruction is written as a separate statement. The C program starts with a main function followed by the opening braces which indicates the start of the function. Then follows the variable and constant declarations which are followed by the statements that include input and output statements.

C program may contain one or more sections as shown below:

DOCUMENTATION SECTION

LINK SECTION

DEFINITION SECTION

GLOBAL DECLARATION SECTION

Main() Function section

{

Declaration part

Executable part

}

SUBPROGRAM SECTION

User defined functions

Keywords

C has 32 keywords (reserved words with special meaning):

1. auto,
 2. break,
 3. case,
 4. char,
 5. const,
 6. continue,
 7. default,
 8. do,
 9. double,
 10. else,
 11. enum,
 12. extern,
 13. float,
 14. for,
 15. goto,
 16. if,
 17. int,
 18. long,
 19. register,
 20. return,
 21. short,
 22. signed,
 23. sizeof,
 24. static,
 25. struct,
 26. switch,
 27. typedef,
 28. union,
 29. unsigned,
 30. void,
 31. volatile, and
 32. while.
-

Operators

C supports a rich set of operators, which are symbols used within an expression to specify the manipulations to be performed while evaluating that expression. C has the following operators:

arithmetic: +, -, *, /, %
assignment: =
augmented assignment: +=, -=, *=, /=, %=, &=, |=, ^=, <<=, >>=
bitwise logic: ~, &, |, ^
bitwise shifts: <<, >>
boolean logic: !, &&, ||
conditional evaluation: ? :
equality testing: ==, !=
calling functions: ()
increment and decrement: ++ and --
member selection: ., ->
object size: sizeof
order relations: <, <=, >, >=
reference and dereference: &, *, []
sequencing: ,
subexpression grouping: ()
type conversion: (*typename*)

SALIENT FEATURES OF C

C language has some characteristics that define the language and also have led to its popularity as a programming language.

- Small size.
 - Extensive use of function calls.
 - Structured language.
 - Low level (Bitwise) programming readily available.
 - Pointer implementation - extensive use of pointers for memory, array, structures and functions.
 - It has high-level constructs.
 - It can handle low-level activities.
 - It produces efficient programs.
 - It can be compiled on a variety of computers.
-

INSTRUCTIONS TO STUDENTS FOR PREPARING PROGRAMMING IN C LAB REPORT

This Lab Manual is prepared to help the students with their practical understanding and development of programming skills, and may be used as a base reference during the lab/practical classes.

Students have to submit Lab Exercise report of previous lab into corresponding next lab, and can be collected back after the instructor/course co-ordinator after it has been checked and signed. At the end of the semester, students should compile all the Lab Exercise reports into a single report and submit during the end semester sessional examination.

“**Sample of Lab report**” is shown for LAB Exercise #1 in this manual. For the rest of the labs, the reporting style as provided is to be followed. The lab report to be submitted during the end semester Sessional Examination should include at least the following topics:-

1. Cover page – *(to be attached on top of the Lab Exercise)*
2. Index – *(to be used while compiling all the Lab Exercise reports into single report)*
3. Title of the program
4. Algorithm
5. Flowchart *(optional)*
6. Coding
7. Output (compilation, debugging & testing)

Students are to show the coding and output to the instructor/course co-ordinator for the additional lab exercises given in each lab module.

Note: The lab exercises may not be completed in a single specific lab. Students are encouraged to complete the programming questions given in the exercise prior to come to the lab hour and do the lab for the given programs.

LAB EXERCISE #1

Objective(s):

To be familiar with syntax and structure of C- programming.
To learn problem solving techniques using C

Program: Write a Program to calculate and display the volume of a CUBE having its height (h=10cm), width (w=12cm) and depth (8cm).

Algorithm:

1. Start
2. Define variables: h(int), w(int), d(int), vol(int)
3. Assign value to variables: h = 10, w=12, d=8
4. Calculate the volume as: $vol = h*w*d$
5. Display the volume (vol)
6. Stop

Code: *(Use comments wherever applicable)*

```
//Following code is written and compiled in GCC

#include<stdio.h>
void main()
{
//start the program
int h,w,d,vol; //variables declaration
h=10;w=12;d=8; //assign value to variables
vol=h*w*d; //calculation using mathematical formula
printf("The Volume of the cube is: %d",vol); //display the
volume
getch();
//end the main program
}
```

Output :

The Volume of the cube is: 960

SAMPLE PROGRAMS

(Students are to code the following programs in the lab and show the output to instructor/course co-ordinator)

Instructions

- *Write comment to make your programs readable.*
- *Use descriptive variables in your programs(Name of the variables should show their purposes)*

Programs List

1. Write a C program to display “This is my first C Program”.
 2. Write a C program to add two numbers (2 and 6) and display its sum.
 3. Write a C program to multiply two numbers (4 and 5) and display its product.
 4. Write a C program to calculate area and circumference of a circle.
 5. Write a C program to perform addition, subtraction, division and multiplication of two numbers.
 6. Write C program to evaluate each of the following equations.
(i) $V = u + at$. (ii) $S = ut + \frac{1}{2}at^2$ (iii) $T = 2\sqrt{a + \sqrt{b + 9c}}$ (iv) $H = \sqrt{b^2 + p^2}$
-
-

LAB EXERCISE #2

Objective(s):

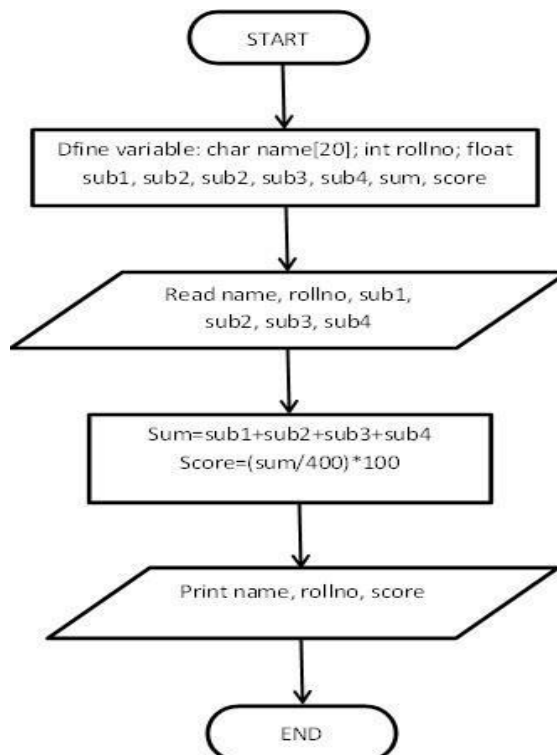
To be familiar with different data types, Operators and Expressions in C.

Program: Write a program to take input of name, rollno and marks obtained by a student in 4 subjects of 100 marks each and display the name, rollno with percentage score secured.

Algorithm:

1. Start
2. Define variables: name, rollno, sub1, sub2, sub3, sub4, sum, score
3. Take input from keyboard for all the input variables
4. Calculate the sum of marks of 4 subjects and also calculate the percentage score as:
 $sum = sub1 + sub2 + sub3 + sub4;$
 $score = (sum/400) * 100$
5. Display the name, roll number and percentage score.
6. Stop

Flowchart:



Code: *(Use comments wherever applicable)*

//Following code is written and compiled in TURBO C++

```
#include<stdio.h>
#include<conio.h>
void main()
{
char name[20];
int rollno;
float sub1, sub2, sub3, sub4, , sum, score;
printf("Enter name of student: ");
scanf("%s",&name[]);
printf ("\n Enter Roll Number: ");
scanf("%d", &rollno);
printf ("\n Enter Marks in 4 Subjects:\n");
scanf("%f%f%f%f", &sub1, &sub2, &sub3, &sub4);
sum=sub1+sub2+sub3+sub4;
score = (sum/500)*100;
printf("\n Name of student: %s", name[]);
printf("\n Roll Number: %d", rollno);
printf ("\nPercentage score secured: %2.2f%c", score,'%');
getch();
}
```

Output:

Enter name of student: Ajit Singh

Roll Number: 25

Enter Marks in 4 Subjects:

50

75

85

62

Name of student: Ajit Singh

Roll Number: 25

Percentage score secured: 68.00%

SAMPLE PROGRAMS

(Students are to code the following programs in the lab and show the output to instructor/course co-ordinator)

Instructions

- *Write comment to make your programs readable.*
- *Use descriptive variables in your programs(Name of the variables should show their purposes)*

Programs List

1. Write a program to calculate simple and compound interest.
 2. Write a program to swap values of two variables with and without using third variable.
 3. Write a program to display the size of every data type using “sizeof” operator.
 4. Write a program to illustrate the use of unary prefix and postfix increment and decrement operators.
 5. Write a program to input two numbers and display the maximum number.
 6. Write a program to find the largest of three numbers using ternary operators.
 7. Write a program to find the roots of quadratic equation.
 8. Write a program to input name, marks of 5 subjects of a student and display the name of the student, the total marks scored, percentage scored and the class of result.
-

LAB EXERCISE #3

Objective(s):

To understand the programming knowledge using Decision Statements (if, if-else, if-else-if ladder, switch and GOTO)

Program: Write a program to print whether a given number is even or odd.

Code: *(Use comments wherever applicable)*

//Following code is written and compiled in TURBO C++

```
#include<stdio.h>
#include<conio.h>
void main()
{
int num;
printf("Enter the number: ");
scanf("%d",&num);
if(num%2==0)
    printf("\n %d is even", num);
else
    printf("\n %d is odd", num);
getch();
}
```

Output:

Enter the number: 6
6 is even

SAMPLE PROGRAMS

(Students are to code the following programs in the lab and show the output to instructor/course co-ordinator)

Instructions

- *Write comment to make your programs readable.*
- *Use descriptive variables in your programs(Name of the variables should show their purposes)*

Programs List

1. Write a Program to Check Whether a Number is Prime or not.
2. Write a program to find the largest and smallest among three entered numbers and also display whether the identified largest/smallest number is even or odd.
3. Write a program to compute grade of students using **if else adder**. The grades are assigned as followed:

<u>Marks</u>	<u>Grade</u>
b. $\text{marks} < 50$	F
c. $50 \leq \text{marks} < 60$	C
d. $60 \leq \text{marks} < 70$	B
e. $70 \leq \text{marks} < 80$	B+
f. $80 \leq \text{marks} < 90$	A
g. $90 \leq \text{mars} \leq 100$	A+

4. Write a program to check whether the entered year is leap year or not (a year is leap if it is divisible by 4 and divisible by 100 or 400.)
 5. Write a program to find the factorial of a number.
 6. Write a program to check number is Armstrong or not.
(Hint: A number is Armstrong if the sum of cubes of individual digits of a number is equal to the number itself).
-

Program: Write a program to find whether a character is consonant or vowel using **switch statement**.

```
#include <stdio.h>
void main()
{
char ch;
printf("Enter any alphabet:"); //input alphabet from user
scanf("%c", &ch);
switch(ch)
{
    case „a“:
    case „A“:
        printf("Vowel");
        break;
    case „e“:
    case „E“:
        printf("Vowel");
        break;
    case „i“:
    case „I“:
        printf("Vowel");
        break;
    case „o“:
    case „O“:
        printf("Vowel");
        break;
    case „u“:
    case „U“:
        printf("Vowel");
        break;
    default:
        printf("Consonant");
}
}
```

7. Write a program to print day name using **switch case**.
 8. Write a program to determine whether the input character is capital or small letter, digits or special symbol.
 9. Write a program to check whether a date is valid or not.
 10. Write a program to check whether a number is positive, negative or zero using switch case.
-

LAB EXERCISE #4

Objective(s):

To understand the programming using Loop & nested loop Statements (for, while, do-while)

Program: Write a program to print positive integers from 1 to 10.

Code:

```
//Following code is written and compiled in TURBO C++  
//Using FOR LOOP
```

```
#include<stdio.h>  
#include<conio.h>  
void main()  
{  
int i;  
for(i=1; i<=10;i++)  
    printf("%d \n", i);  
getch();  
}
```

```
//Using WHILE LOOP
```

```
#include<stdio.h>  
#include<conio.h>  
void main()  
{  
int i=1;  
while(i<=10)  
    {  
    printf("%d \n", i);  
    }  
    i++;  
getch();  
}
```

```
//Using DO-WHILE LOOP

#include<stdio.h>
#include<conio.h>
void main()
{
int i=1;
do
    {
    printf("%d \n", i);
    i++;
    }
while(i<=10);
getch();
}
```

Output:

1
2
3
4
5
6
7
8
9
10

SAMPLE PROGRAMS

(Students are to code the following programs in the lab and show the output to instructor/course co-ordinator)

Instructions

- Write comment to make your programs readable.
- Use descriptive variables in your programs (Name of the variables should show their purposes)

Programs List

1. Write a program to count number of digits in a given integer.
2. Write a program to reverse a given integer.
3. Write a program to print number in reverse order with a difference of 2.
4. Write a program to print the sum of digits of a number using **for** loop.
5. Write a program to check whether a number is Palindrome or not.
6. Write a program to generate Fibonacci series.
7. If a four-digit number is input through the keyboard, write a program to obtain the sum of the first and last digit of this number.
8. Write a program to find GCD (greatest common divisor or HCF) and LCM (least common multiple) of two numbers.

Program: Write a program to display the following pattern.

```
*
* *
* * *
* * * *
* * * * *
```

Code:

```
#include<stdio.h>
#include<conio.h>
void main()
{
int i,j;
for(i=1; i<=5;i++)
{
for(j=1;j<=i;j++)
{
printf("*");
}
printf("\n");
}
getch();
}
```

9. Write programs to display each of the following patterns.

(i)
* * * * *
* * * *
* * *
* *
*

(ii)
1
2 2
3 3 3
4 4 4 4
5 5 5 5 5

(iii)
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5

(iv)
A
A B
A B C
A B C D
A B C D E

(v)
*
* * *
* * * * *
* * * * * * *
* * * * * * * * *

(vi)
* * * * * * * * * *
* * * * * * * *
* * * * * *
* * * *
* * *
*

(vii)
1
1 2 1
1 2 3 2 1
1 2 3 4 3 2 1
1 2 3 4 5 4 3 2 1

(viii)
A B C D E F
A B C D E
A B C D
A B C
A B
A

(ix)
1
1 2 3
1 2 3 4 5
1 2 3
1

(x)
* * * * * * * *
* * * * * * *
* * * * * *
* * * * *
* * * * *

(xi)
* * * * * * *
* * * * * *
* * * * * *
* * * * * *
* * * * * *
* * * * * *
* * * * * *
* * * * * *
* * * * * *

(xii)
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *



LAB EXERCISE #5

Objective(s):

To understand programming using different dimensions of Array.

Program: Write a program to insert 5 elements into an array and print the elements of the array.

Code: *(Use comments wherever applicable)*

```
#include<stdio.h>
#include<conio.h>
void main()
{
int i, arr[5];
printf("Enter the elements into the array:");
for(i=0; i<=4;i++)
    scanf("%d",&arr[i]);
printf("The elements of the array are:");
for(i=0; i<=4;i++)
    printf("%d \t", arr[i]);
getch();
}
```

SAMPLE PROGRAMS

(Students are to code the following programs in the lab and show the output to instructor/course co-ordinator)

Instructions

- Write comment to make your programs readable.
- Use descriptive variables in your programs(Name of the variables should show their purposes)

Programs List

1. Write a Program to Search an element in array.
 2. Write a Program to perform addition of all elements in Array.
 3. Write a Program to find the largest and smallest element in Array.
-

4. Write a Program to reverse the array elements in C Programming.
 5. Write a Program for deletion of an element from the specified location from Array.
 6. Write a Program to access an element in 2-D Array.
 7. Write a program for addition of two matrices of any order in C.
 8. Write a Program to multiply two 3 X 3 Matrices.
 9. Write a program to read a string and check for palindrome without using string related function (a string is palindrome if its half is mirror by itself eg: abcdcba).
 10. Write a program to accept a string and count the number of vowels present in this string.
-

LAB EXERCISE #6

Objective(s):

To understand function programming, its types and function-call.

Program: Write a program to calculate factorial of a number using recursion.

Code:

```
#include<stdio.h>
long factorial(int); //Function declaration
int main()
{
    int num;
    long fact;
    printf("Enter a number to find factorial: \n");
    scanf("%d", &num);
    if(num<0)
        printf("Factorial of negative no. is not defined. \n");
    else
    {
        fact = factorial(num);
        printf("%d!=%d \n", num, fact);
    }
    return 0;
}
//Function definition
long factorial(int num)
{
    if(num==0)
        return 1;
    else
        return(num*factorial(num-1));
}
```

SAMPLE PROGRAMS

(Students are to code the following programs in the lab and show the output to instructor/course co-ordinator)

Instructions

- *Write comment to make your programs readable.*
- *Use descriptive variables in your programs(Name of the variables should show their purposes)*

Programs List

1. Write a program to add, subtract, multiply and divide two integers using user-defined type function with return type.
 2. Write a program to calculate sum of first 20 natural numbers using recursive function.
 3. Write a program to generate Fibonacci series using recursive function.
 4. Write a program to swap two integers using call by value and call by reference methods of passing arguments to a function.
 5. Write a program to find sum of digits of the number using Recursive Function.
 6. Write a program to read an integer number and print the reverse of that number using recursion.
 7. Write a C program to find maximum and minimum between two numbers using functions.
 8. Write a C program to check whether a number is even or odd using functions.
 9. Write a C program to check whether a number is prime, Armstrong or perfect number using functions.
 10. Write a C program to find power of any number using recursion.
-

LAB EXERCISE #7

Objective(s):

To understand programming with Pointer, String and Function call by reference.

Program: Write a program to find biggest among three numbers using pointer.

Code:

```
#include<stdio.h>
#include<conio.h>
int main()
{
int a,b,c;
int*ptr a=&a,*ptr b=&b,*ptr c=&c;
printf("enter three values");
scanf("%d%d%d",ptr a,ptr b,ptr c);
printf("a=%d\n b=%d\n c=%d\n",*ptr a,*ptr b,*ptr c);
if((*ptr a>*ptr b && *ptr a>*ptr c)
    printf("biggest number=%d",*ptr a);
else if((*ptr b>*ptr a && *ptr b>*ptr c)
    printf("biggest number =%d",*ptr b);
else
    printf("biggest number=%d",*ptr c);
getch();
return 0;
}
```

SAMPLE PROGRAMS

(Students are to code the following programs in the lab and show the output to instructor/course co-ordinator)

Instructions

- *Write comment to make your programs readable.*
- *Use descriptive variables in your programs(Name of the variables should show their purposes)*

Programs List

1. Write a program to find the sum of all the elements of an array using pointers.
 2. Write a program to swap value of two variables using pointer.
 3. Write a program to add two numbers using pointers.
 4. Write a program to input and print array elements using pointer.
 5. Write a program to copy one array to another using pointer.
 6. Write a program to swap two arrays using pointers.
 7. Write a program to reverse an array using pointers.
 8. Write a program to search an element in array using pointers.
 9. Write a program to add two 2 X 2 matrix using pointers.
 10. Write a program to multiply two 2 X 2 matrix using pointers.
 11. Write a program to find length of string using pointers.
 12. Write a program to copy one string to another using pointer.
 13. Write a program to concatenate two strings using pointers.
 14. Write a program to compare two strings using pointers.
-

LAB EXERCISE #8

Objective(s):

To understand programming with Structure.

Program 1: Write a C program to create, declare and initialize structure.

Code:

```
#include <stdio.h>
/*structure declaration*/
struct employee{
    char name[30];
    int empId;
    float salary;
};

int main()
{
    /*declare and initialization of structure variable*/
    struct employee emp={"Anil",201,80000.00};

    printf("\n Name: %s"        ,emp.name);
    printf("\n Id: %d"         ,emp.empId);
    printf("\n Salary: %f\n",emp.salary);
    return 0;
}
```

Program 2: Write a program to store information of 5 students in structure and display it.

Code:

```
#include<stdio.h>
struct student
{
    char name[30];
    int roll;
    float marks;
} s[5];
int main( )
{
    int i;
    printf("Information of students:");
```

```

for (i=0; i<5; ++i)
{
    s[i].roll =i+1;
    printf("\n Roll number %d, \n", s[i].roll);
    printf("Enter name:");
    scanf("%s", s[i].name);
    printf("Enter marks:");
    scanf("%f", &s[i].marks);
}
printf("\n Displaying Information:\n");
for(i=0;i<10;++i)
{
    printf("\n Roll number:%d \n", i+1);
    printf("Name:");
    puts(s[i].name);
    printf("\n Marks:%.1f", s[i].marks);
}
return 0;
}

```

Program 3: Write a program to declare, initialize an UNION.

Code:

```

#include <stdio.h>
// union declaration
union pack{
char a;
int b;
double c;
};
int main()
{
    pack p; //union object/variable declaration
    printf("\nOccupied size by union pack:
%d",sizeof(pack));
    // assign value to each member one by one other it
will replace last value
    p.a='A';
    printf("\nValue of a:%c",p.a);
    p.b=10;
    printf("\nValue of b:%d",p.b);
    p.c=12345.6790;
    printf("\nValue of c:%f",p.c);
}

```

```

    // see, what will happen? if u will assign values
together
    p.a='A';
    p.b=10;
    p.c=12345.6790;
    // here the last value of p.c will be accessed by all
members
    printf("\nValue of a:%c, b:%d, c:%f",p.a,p.b,p.c);
    return 0;
}

```

SAMPLE PROGRAMS

(Students are to code the following programs in the lab and show the output to instructor/course co-ordinator)

Instructions

- *Write comment to make your programs readable.*
 - *Use descriptive variables in your programs(Name of the variables should show their purposes)*
1. Write a program to create a structure named company which has name, address, phone and noOfEmployee as member variables. Read name of company, its address, phone and noOfEmployee. Finally display these members" value.
 2. Define a structure "complex" (typedef) to read two complex numbers and perform addition, subtraction of these two complex numbers and display the result.
 3. Write a program to read RollNo, Name, Address, Age & average-marks of 12 students in the BCT class and display the details from function.
 4. Write a program to add two distances in feet and inches using structure
 5. Write a program to read and print an Employee"s Details using Structure.
-

LAB EXERCISE #9

Objective(s):

To understand data files and file handling in C.

Program 1: Write a program to create a file called emp.rec and store information about a person, in terms of his name, age and salary.

Code:

```
#include <stdio.h>
void main()
{
    FILE *fptr;
    char name[20];
    int age;
    float salary;
    /* open for writing */
    fptr = fopen("emp.rec", "w");
    if (fptr == NULL)
    {
        printf("File does not exists \n");
        return;
    }
    printf("Enter the name \n");
    scanf("%s", name);
    fprintf(fptr, "Name      = %s\n", name);
    printf("Enter the age\n");
    scanf("%d", &age);
    fprintf(fptr, "Age        = %d\n", age);
    printf("Enter the salary\n");
    scanf("%f", &salary);
    fprintf(fptr, "Salary   = %.2f\n", salary);
    fclose(fptr);
}
```

Program 2: Write a program to illustrate how a file stored on the disk is read.

Code:

```
#include <stdio.h>
#include <stdlib.h>
void main()
{
    FILE *fptr;
    char filename[15];
    char ch;
    printf("Enter the filename to be opened \n");
    scanf("%s", filename);
    /* open the file for reading */
    fptr = fopen(filename, "r");
    if (fptr == NULL)
    {
        printf("Cannot open file \n");
        exit(0);
    }
    ch = fgetc(fptr);
    while (ch != EOF)
    {
        printf ("%c", ch);
        ch = fgetc(fptr);
    }
    fclose(fptr);
}
```

SAMPLE PROGRAMS

(Students are to code the following programs in the lab and show the output to instructor/course co-ordinator)

Instructions

- *Write comment to make your programs readable.*
 - *Use descriptive variables in your programs(Name of the variables should show their purposes)*
1. C Program to list all files and sub-directories in a directory
 2. C Program to count number of lines in a file
 3. C Program to print contents of file
 4. C Program to copy contents of one file to another file
 5. C Program to merge contents of two files into a third file
 6. C program to delete a file
-