Expt. No. 4 2) Page No. 36 Date

2) Declare a local variable inside a function and try to access it outside the function. Compare this with accessing the global variable from within the function.

```
#include <stdio.h>
  int a=6;
  int sum()
  {
  int b=4;
  return a+b;
  }
  int main()
  {
  printf("a:%d\t b:%d", a,b);
  printf("%d", sum());

  return 0;
  }
```

Teacher's Signature

Output - Error because b is a local variable of Sum function and we are trying to access it outside Sum function are being a global variable can be accessed anywhere.

Expt. No. 4 3) Page No. 37 Date

3) Declare variables within different code blocks (enclosed by curly braces) and test their accessibility within and outside those blocks.

```
#include <stdio.h>
int sum();
int sub();

int main()
{
printf("%d %d %d %d", a, b, c, d);

printf("%d \n %d", sum(), sub());

return 0;
}
int sum()
{
int a=6, b=4;
return a+b;
}
int sub()
{
int c=5, d=4;
return c-d;
}
```

Teacher's Signature

Output :-

Because a, b, c, d are local variables of Sum and Sub functions. So they can be accessed only within those functions and not outside them.

Expt. No. 4 4) Page No. 38 Date

4) Declare a static local variable inside a fn. Observe how its value persists across function calls.

```
#include <stdio.h>
int sum()
{
static int a=0;
int sum = a+5;
a= sum;
printf("%d\n", sum);
return a;
}
int main()
{
  sum();
  sum();
  sum();
  sum();

return 0;
}
```

Teacher's Signature

Output :-

5
10
15
20

If a normal variable is used then it gets destroyed as soon as function terminates so if we had used a normal variable then output would have been:

5
5
5
5

If a static variable is formed then value of variable persists between function calls

Expt. No. 7 1) Page No. 39 Date

EXPERIMENT - 7

1) Write a program that uses function to perform the operations.

```
#include <stdio.h>
struct complex
  int rp, ip;
};
struct complex read()
{
struct complex c;
printf("Enter the real part ");
scanf("%d", &c.rp);
printf("Enter the imaginary part:");
scanf("%d", &c.ip);
  return c;
}
void write (struct complex c)
{ printf("%d + %di \n", c.rp, c.ip);
}
struct complex add (struct complex a, struct
                    complex b)
{
```

Teacher's Signature

Expt. No. 7 1) Page No. 40 Date

```
struct complex res;
 res.rp = a.rp + b.rp;
res.ip = a.ip + b.ip;
 return res;
}
struct complex sub (struct complex a, struct complex b)
{ struct complex res;
res.rp = a.rp - b.rp;
res.ip = a.ip - b.ip;
 return res;
}
int main()
{
struct complex c1, c2, c3, c4;
printf ("Enter first complex number");
 c1 = read();
print ("Enter second complex number");
c2 = read();
 write (c1);
 printf("\n");
 write (c2);
c3 = add (c1, c2)
c4 = sub (c1, c2)
print ("\n");
 write (c3);
 write (c4);
 return 0;
}
```

Teacher's Signature

Expt. No. 7 2) Page No. 41 Date

2) W.AP to compute the monthly pay of 100 employees using each employees name, basic pay. the DA is computed as 52% of basic pay. Gross salary.

```
#include<stdio.h>
struct employee
{
 char name[100];
  int basic;
  float gross;
};
int main()
{
struct employee emp[100];
for(int i=0; i<100; i++)
{
printf("Employee %d\n", i+1);
 printf("Enter name:");
 Scanf("%s", emp[i].name);
printf("Enter basic pay");
Scanf("%d", &emp[i].basic);
float DA = 0.5) emp[i].basic;
   emp[i].gross = emp[i].basic + DA
}
 printf("\n")
```

Teacher's Signature

OUTPUT → Employee - 1
Enter name : Riya
Enter basic Pay : 10000

Employee - 2
Enter name : Arjun
Enter basic Pay : 15000

Employee 3
Enter name : Meera
Enter basic Pay : 20000

Employee1: Name : Riya, Gross Salary : 15000
Employee 2 : Name : Arjun, Gross Salary : 22500
Employee 3 : Name : Meera, Gross Salary : 30000

```
for (int i =0; i<100; i++)
  {
printf ("%.5s; %.0.2f \n", emp[i].name, emp[i].gross);
  }
return 0;
}
```

INDEX

S. No.	Name of the Experiment	Page No.	Date of Experiment	Date of Submission	Rem
7)	Structure and Union	1-3	11/9/25		
9)	File Handling in C	4-6	25/9/25		
10)	Dynamic Memory Allocation	7-11	9/10/25		
12)	Preprocessor and Directives in C	12,13	16/10/25		
13)	Macros in C	14	23/10/25		
14)	Static Library in C	15,17	30/10/25		
15)	Shared Library in C	18,19	6/11/25		

OUTPUT :- Enter book id: 101
Enter book title : C programming
Enter book author : Dennis Ritchie
Enter book price : 450

ID : 101
TITLE : C programming
Author : Dennis Ritchie
Price : 450

Expt. No. 7 3) Page No. 1 Date

EXPERIMENT - 7

3) Create a book struct containing book_id, title, author name and price. write a C program to pass a structure as a function ---- details

```
struct book
{
int id, price;
char title[50], auther[50];
};
void details (struct book a)
{
printf ("id : %d \n", a.id);
printf ("Title : %s \n", a.title);
printf ("Auther : %s \n", a.auther);
printf ("Price : %d \n", a.price);
}
int main()
{
struct book a;
printf ("Enter book id : ");
scanf ("%d", &a.id);
printf ("Enter book title");
scanf ("%s", a.title);
```

Teacher's Signature

```
printf("Enter Author's name");
scanf("%s", a.author);

printf("Enter book price:");
scanf("%d", &a.price);

details (a);
  return 0;
}
```

Teacher's Signature

Expt. No. 7 4) Page No. 3 Date

4) Create a union containing 6 strings: name, home address, hostel address, city, state and zip. write a C program to display your present address.

```
#include <stdio.h>
  union address
  {
  char name[50];
  char home_address[100], hostel_address[100],
  city[50], state[50], zip[10];
    };

int main()
  {
  union address a;
  printf("Enter your present hostel address: ");
  fgets(a.hostel_address, size of(a.hostel_address),
                                    stdin);
  printf("Your present address is: ");
  printf("%s", a.hostel_address);
  return 0;
  }
```

Teacher's Signature

OUTPUT :- Enter your present hostel address:
Room 214, ABC Girls Hostel, Chandigarh

Your present address is:
Room 214, ABC Girls Hostel, Chandigarh

Expt. No. a i) 2) Page No. 4 Date

EXPERIMENT - 9

1) WAP to create a new file and write text into it.

```
#include <stdio.h>
int main()
{
FILE * ptr;
ptr= fopen ("file.text", "w");
fprintf (ptr, "Hello everyone how are you");
fclose (ptr);
printf("File is created");
return 0;
}
```

2) Open an existing file and read its content character by character, and then close the file.

```
#include <stdio.h>
int main()
{
FILE * ptr;
char ch;
ptr= fopen ("test.txt", "r");
```

Teacher's Signature

OUTPUT:- File is created.

OUTPUT:- Hello everyone how are you

```
if (ptr == NULL)
{
  printf ("FILE doesnot exist");
    return 1;
}
while ((ch = fgetc (ptr)) != EOF)
{
  printf ("%c", ch);
}
  fclose (ptr);
  return 0;
  }
```

Expt. No. 9 3) Page No. 6 Date

3) Open a file read its content line by line, and display each line on the console.

```
#include<stdio.h>
int main()
{
FILE *ptr;
char a[100];
ptr=fopen("test.txt","r");
if(ptr==NULL)
{printf("FILE Doesnot exist");
return 1;}
while(fgets(a,sizeof(a),ptr)!=NULL)
{
printf("%s",a);
}
fclose(ptr);
return 0;
}
```

Teacher's Signature

OUTPUT :

```
Hello everyone
This is a file test
C programming is easy
```

Expt. No. 10) Page No. 7 Date

EXPERIMENT -10

1) WAP to create a simple linked list in C using pointer and structure

```
#include <stdio.h>
#include <stdlib.h>
struct node{
int data;
struct node* next;
};
int main
{
struct node *first, *second, *third;
first = (struct node *) malloc (size of (struct node));
second = (struct node *) malloc (size of (struct node));
third = (struct node*) malloc (size of (struct node));

first -> data = 10;
first -> next = second;
second -> data = 20;
second -> next = third;
third -> data = 30;
third -> next = NULL;
struct node * temp = first;
```

Teacher's Signature

(3)

OUTPUT - 10 20 30

```
while (temp != NULL)
{
printf("%d →", temp → data);
    temp = temp → next;
}
return 0;
}
```