Experiment No. 9 Name: _____ PAGE NO.: DATE: / /

EXPERIMENT : 9

FILE HANDLING IN C

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

```
#include <stdio.h>
int main () {
  char text [200] ;

  printf ("Enter text : \n") ;
  fgets (text , sizeof (text ), stdin );

  printf ("Pretend writing to file : %s\n", text);
  printf ("File created successfully (simulation
                    only) .\n");

  return 0;
}
```

Output :- Hello Everyone I am studing in upes university dehradun. Dehradun is the very beautiful place.
Pretend writing to file : Hello everyone I am studing in upes university dehradun. Dehradun is the very beautiful place.

Teacher's Signature: _____

main.c

Share Run

```c
#include <stdio.h>

int main() {
    FILE *fp;
    char text[100];

    // Open file in write mode
    fp = fopen("newfile.txt", "w");

    // Check if file is created successfully
    if (fp == NULL) {
        printf("Error! Unable to create file.\n");
        return 1;
    }

    printf("Enter text to write into the file:\n");
    fgets(text, sizeof(text), stdin);

    // Write the text into the file
    fputs(text, fp);

    printf("Text written successfully into 'newfile.txt'\n");

    // Close the file
    fclose(fp);

    return 0;
}
```

Output

Clear

```
Error! Unable to create file.


=== Code Exited With Errors ===
```

Experiment No. _____ Name: _____

2. Write a C program to compute the monthly pay of 100 employees using each employee's name, basic pay. The DA is computed as 52% of the basic pay. Gross - salary (basic pay + DA). Print the employees name and gross salary.

```
#include <stdio.h>
# define EMP_ COUNT 10

struct Employee {
        char name [50] ;
        float basic_pay ;
        float da ;
        float gross_salary ;
} ;
int main ( ) {
  struct Employee emp [EMP_COUNT ] ;
  int i ;

  printf ("Enter details of %d employees : \n",
                                  EMP_ COUNT ) ;
  for (i = 0 ; i < EMP_ COUNT ; i++ ) {
      printf ("\n Employee %d \n", i+1) ;
      printf ("Enter name : ") ;
      scanf ("%s", emp [i]. name) ;
      printf ("Enter basic pay : ") ;
      scanf ("%f", & temp [i]. basic_ pay) ;
```

Teacher's Signature: _____

```
    emp[i].da = 0.52f * emp[i].basic_pay;
    emp[i].gross_salary = emp[i].basic_pay +
                          emp[i].da;
}
printf("\n%-20s%-15s\n", "Employee
        Name", "Gross Salary");
printf("\n%-20s%-15s\n", "Employee
        Name", "Gross Salary\n");
for(i=0; i<EMP_COUNT; i++) {
    printf("%-20sRs. %.2f\n", emp[i].name,
           emp[i].gross_salary);
}
return 0;
}
```

main.c

Share Run

```c
#include <stdio.h>

int main() {
    FILE *fp;
    char ch;

    fp = fopen("newfile.txt", "r");

    if (fp == NULL) {
        printf("Error! File not found.\n");
        return 1;
    }

    printf("Contents of the file:\n\n");

    while ((ch = fgetc(fp)) != EOF) {
        putchar(ch);
    }

    fclose(fp);

    printf("\n\nFile read successfully and closed.\n");

    return 0;
}
```

Output Clear

```
Error! File not found.


=== Code Exited With Errors ===
```

3). Write a function that accept pointers as parameters. Pass variables by reference using pointers and modify their values within the function.

```
#include <stdio.h>

void modifyValues(int *x, float *y, char *z){
    *x = *x + 10;
    *y = *y + 2;
    *z = *z + 1;
}
int main() {
    int a = 5;
    float b = 3.5;
    char c = 'A';

    printf("Before modification:\n");
    printf("a = %d\n", a);
    printf("b = %.2f\n", b);
    printf("c = %c\n\n", c);

    modifyValues(&a, &b, &c);

    printf("After modification (inside function):\n");
    printf("a = %d\n", a);
    printf("b = %.2f\n", b);
    printf("c = %c\n\n", c);
```

Teacher's Signature: _____

```
    return 0;
}
```

Output :- Before modification :

a = 5

b = 3.50

c = A

After modification (inside function) :

a = 15

b = 7.00

c = B

main.c

Share Run

```c
#include <stdio.h>

int main() {
    FILE *fp;
    char line[200];

    fp = fopen("newfile.txt", "r");

    if (fp == NULL) {
        printf("Error! File not found.\n");
        return 1;
    }

    printf("Contents of the file:\n\n");

    while (fgets(line, sizeof(line), fp) != NULL) {
        printf("%s", line);
    }

    fclose(fp);

    printf("\n\nFile read successfully and closed.\n");

    return 0;
}
```

Output Clear

```
Error! File not found.


=== Code Exited With Errors ===
```