

Theory Assignment – 05

```
// 1. Write a C program to read an existing file.

#include <stdio.h>

int main(){
    FILE * file;

    file = fopen("text.txt", "r");
    char line[100];

    while (fgets(line, 100, file))
    {
        printf("%s", line);
    }

    fclose(file);
    return 0;
}
```

Output :

Hello World
Welcome in C

Q2. Explain file handling through command line argument.

Ans:

File handling is an essential aspect of programming in C language. In C, files can be created, opened, read, written, and closed using functions provided by the standard library. Command line arguments can be used to pass the name of the file to be processed to the program.

Here is an example of how file handling can be done through command line arguments in C language:

```
#include <stdio.h>

int main(int argc, char *argv[]) {
    FILE *fp;
    char ch;

    // Check if a filename was provided on the command line
    if (argc < 2) {
        printf("Usage: %s filename\n", argv[0]);
        return 1;
    }

    fp = fopen(argv[1], "r");

    while ((ch = fgetc(fp)) != EOF) {
        printf("%c", ch);
    }
    fclose(fp);

    return 0;
}
```

c:\TA - 5> program.exe text.txt

Hello World
Welcome in C

3. How do we declare a file in a C program? Explain the opening modes of file.

```
#include <stdio.h>

int main(){
    FILE * file; // Declaration of file pointer

    file = fopen("text.txt", "r"); // Opening file in read mode
    fclose(file); // Closing the file

    fclose(file);
    return 0;
}
```

Different file modes

- r – Opens the file in read mode only.
- w – Opens the file in write mode only.
 - Opens the file with no previous data.
- a – Opens the file in append mode.
 - Opens the file and appends more data in existing file.
- a⁺ – Opens the file for both appending and reading.
- r⁺ – Opens the file in read and write mode simultaneously.
- w⁺ – Opens the file in write and read mode simultaneously.

4. What is structure? How can we access element of a structure?

Ans:

In C programming, a structure is a user-defined data type that allows you to group together different types of data under a single name. A structure can contain members of any type, including other structures.

Simple code to access the data present in the structure.

```
#include <stdio.h>

struct Person {
    char name[50];
    int age;
    float height;
};

int main() {
    // Declare and initialize a struct variable
    struct Person p1 = {"John Smith", 35, 1.8};

    // Access the members of the struct variable
    printf("Name: %s\n", p1.name);
    printf("Age: %d\n", p1.age);
    printf("Height: %.2f\n", p1.height);

    return 0;
}
```

5. Write a short note on nested structure in C.

Ans:

In C programming, a nested structure is a structure that contains another structure as one of its members. This allows you to organize related data in a hierarchical manner and access it using a single structure variable.

```
#include <stdio.h>

struct Date {
    int day;
    int month;
    int year;
};

struct Person {
    char name[50];
    struct Date birthdate;
};

int main() {
    struct Person p1 = {"John Smith", {15, 5, 1985}};

    printf("Name: %s\n", p1.name);
    printf("Birthdate: %d/%d/%d\n", p1.birthdate.day, p1.birthdate.month,
p1.birthdate.year);

    return 0;
}
```

In this example, we define two structures: Date and Person. The Date structure has three members: day, month, and year. The Person structure has two members: name, which is a character array, and birthdate, which is a Date structure. We declare and initialize a Person variable named p1 in the main() function, and assign it a name and birthdate. We then use the dot . operator to access the name and birthdate members of the p1 variable and display their values using the printf() function.

Output:

Name: John Smith
Birthdate: 15/5/1985

This shows how a nested structure can be used to group related data and organize it in a hierarchical manner. The Person structure contains a Date structure as one of its members, allowing us to access both the name and birthdate of a person using a single Person variable.