FILES

Read Write

Data Structures and Algorithms

Lab Code: **17ECSP201**

File Structure

```
typedef struct  {
    int level;                     /* fill/empty level of buffer     */
    unsigned  flags;               /* File status flags               */
    char  fd;                      /* File descriptor                 */
     unsigned char hold;           /* Ungetc char if no buffer        */
     int bsize;                    /* Buffer size                     */
     unsigned char  *buffer;       /* Data transfer buffer            */
     unsigned char  *curp;         /* Current active pointer          */
     unsigned  istemp;             /* Temporary file indicator        */
     short token;                  /* Used for validity checking     */
} FILE;
```

To use files, we need to create a variable of type **FILE.**

File Open and Close

```
#include <stdio.h>
int main()
{
        FILE *fp;
        fp = fopen(“file.txt", "w");

        if(fp == NULL)
        {
                printf(“File Open Error!\n");
                return -1;
        }

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

File Write

```
#include <stdio.h>
int main()
{
        char sentence[100];
        FILE *fp;
        fp = fopen(“file.txt", "w");
        if(fp == NULL)  {
                printf(“File Open Error!");
                return -1;
        }
        printf("Enter a sentence with 99 word limit:\n");
        gets(sentence);
        fprintf(fp,"%s", sentence);

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

File Read

```
#include <stdio.h>

int main()
{
        char str[100];
        FILE *fp;
        if ((fp = fopen(“file.txt", "r")) == NULL) {
                printf("Error opening the file\n");
                return -1;
        }

        fscanf(fp,"%[^\n]", str);
        printf("Data from the file:\n%s", str);

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

File Read and Write

```
// read only
FILE *in_file  = fopen("name_of_file", "r");

// write only
FILE *out_file = fopen("name_of_file", "w");

// test for files successful open
if (in_file == NULL || out_file == NULL)
{
        printf("Error! Could not open file\n");
        exit(-1);
}
```

Note: exit() is defined in stdlib.h header file

File Read and Write

```
// write to file vs write to screen

// to file
fprintf(out_file, "this is a test %d\n", integer);


// to screen
fprintf(stdout, "this is a test %d\n", integer);
printf(         "this is a test %d\n", integer);
```

File Read and Write

```
// read from file vs keyboard(console)

// from file
fscanf(in_file, "%d %d", &int_var_1, &int_var_2);



// from console
 fscanf(stdin, "%d %d", &int_var_1, &int_var_2);
 scanf(         "%d %d", &int_var_1, &int_var_2);
```

Thank you.

(More Programs in the Next Lab session)