

Lecture 6

I/O operations on file

Read/Write Operations on Files

- ▶ The simplest file input–output (I/O) functions are **getc** and **putc**.
- ▶ **getc** is used to read a character from a file and return it.

```
char ch; FILE *fp;
```

```
.....
```

```
ch = getc (fp) ;
```

- **getc** will return an end–of–file marker EOF, when the end of the file has been reached.

- ▶ **putc** is used to write a character to a file.

```
char ch; FILE *fp;
```

```
.....
```

```
putc (c, fp) ;
```

Example :: convert a text file to all UPPERCASE

```
main() {  
    FILE *in, *out ;  
    char c ;  
  
    in = fopen ("infile.dat", "r") ;  
    out = fopen ("outfile.dat", "w") ;  
  
    while ((c = getc (in)) != EOF)  
        putc (toupper (c), out);  
    fclose (in) ;  
    fclose (out) ;  
}
```

Contd.

- ▶ We can also use the file versions of **scanf** and **printf**, called **fscanf** and **fprintf**.
- ▶ General format:
`fscanf (file_pointer, control_string, list) ;`
`fprintf (file_pointer, control_string, list) ;`
- ▶ Examples:
`fscanf (fp, "%d %s %f", &roll, dept_code, &cgpa) ;`
`fprintf (out, "\nThe result is: %d", xyz) ;`

Some Points

- ▶ How to check EOF condition when using **fscanf**?

- Use the function **fEOF**

```
if (fEOF (fp))
```

```
    printf (“\n Reached end of file”);
```

- ▶ How to check successful open?

- For opening in “r” mode, the file must exist.

```
if (fp == NULL)
```

```
    printf (“\n Unable to open file”);
```

Example

```
typedef struct {
    int roll;
    char dept_code[6];
    float cgpa;
} STUD;
main() {
    FILE *stud;
    STUD s;
    float sum = 0.0;
    int count = 0;
    stud = fopen ("stud.dat", "r") ;
```

```
while (1) {
    if (feof (stud)) break;
    fscanf (stud, "%d %s %f", &s.roll,
            s.dept_code, &s.cgpa);
    count ++;
    sum += s.cgpa;
}
printf ("\nThe average cgpa is %f",
        sum/count);
fclose (stud);
}
```