Files in C

- Files are used to store data in the storage device.
- We can deal with files in C when we open the file using (open) statement which returns a pointer to the file content.

```
FILE* fp;
fptr = fopen(filename, mode);
if (fptr !=NULL){
      // process the file
}
fflush(fptr);          //OS flushes automatically when buffer is full
fclose(fptr);          //or fcloseall()
```

- There are two different types of files:
  - Text files: stores the data in the form of text.
  - Binary files: stores the data as bytes or as a compiled version.
  - The following table, summaries the most important instructions used with both types:

Text Files	**Binary Files**

Reading:
`fgetc(fptr);`
`fgets(s,100,fptr)`
`fscanf(fptr,"%d",&num);`
	Reading:
	`fread(&x,sizeof(x),1,fptr);`

Writing:
`fputc(ch, fptr);`
`fprintf(fptr, "%s", "Hello");`
	Writing:
	`fwrite(&x,sizeof(x),1,fptr);`

`fseek(fptr, OFFSET, SEEK_END);` `//go to certain location` `printf("%ld\n", ftell(fptr));` `//show the current location`	

Examples-1

```
// read one word at a time from text file
fptr=fopen(filename,"r")
int num;
if(fptr==NULL){
      return 0;
}
while(!feof(fptr){
      fscanf(fptr, "%d", &num);
      printf("%d",num);
}
fclose();
```

Examples-2

```
// read one char at a time (fgetc)
fptr=fopen(filename,"r")
int ch;
if(fptr==NULL){
      return 0;
}
while((ch=fgetc(fptr))!=EOF){
```

```
     printf(“%c”,ch);
}
fclose();
```

Examples-3

```
// write to binary file
int y=10;
int* x=&y;
FILE* fptr= fopen("textfile2.dat","wb");
fwrite(x,sizeof(y),1,fptr);
fclose(fptr);
```

Examples-4

```
// read from binary file
int z;
fptr= fopen("textfile2.dat","rb");
fread(&z,sizeof(y),1,fptr);
fclose(fptr);
printf("%d\n",z);
```