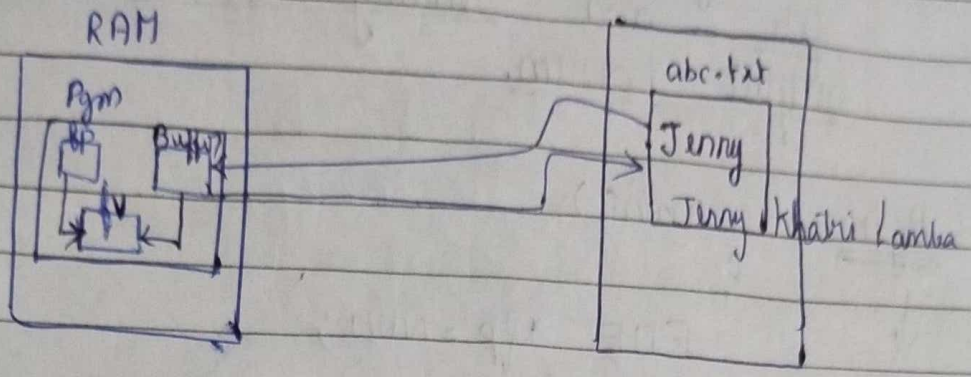
C_122 ⇒ File Handling in C ⇒ Part 4

Read from file in C



* whenever we read the content of a file; the copy of content is file is loaded into buffer and it is read and the content is not overwritten since it is only read mode.

* we can read the file using;
  - → fgetc (read a character
  - → fgets (read a string

NOTE:

↳ If we want to read a file; that specific file should exist in the system and it is a pre requesite or else it will give error

↳ So write a code line to check whether file exist or not means print error for file do not exists.

NOTE:-

* Read mode of file is nothing but accessing a specific data from file and reading it in our output screen.

Pg ①

↓

Read a character from file.

```
main()
{
    FILE *fp = NULL;
    Char ch;
    fp = fopen("abc.txt", "r");
    if (fp == NULL)
    {
        printf("error on file do not exist");
        exit(1);
    }
    ch = fgetc(fp);
    printf("%c", ch);
    fclose(fp);
}
```

O/p :- It will print 'J' on output screen.

Pg ② → Read a character by character (ie) entire string from file (ie) read end of file.

```
main()
{
    FILE *fp = NULL;
```

NOTE:-

feof is predefined function

```
char ch;
fp = fopen("abc.txt", "r");
while(!feof(fp))
{
  ch = fgetc(fp);
  printf("%c", ch);
}
fclose(fp);
}
```

False means 0; it while not go into loop; so put (!) to negative 0 as 1. then it will go into loop

Pg ③ → if we want to read the string directly without character by character.

```
main()
{
  FILE *fp = NULL;
  char str[10];
  fp = fopen("abc.txt", "r");
  if(fp == NULL)
  {
    printf("error or file not exist");
    exit(1);
  }
  fgets(str, 5, fp);
  printf("%s", str);
  fclose(fp);
}
```

here 10 means including null character; So we can print maximum 9 character

max 9 is this case if we give 10

fgets(str, 5, fp); ⇒ 3 arguments

- str → name of string
- 5 → no of character
- fp → file pointer

NOTE

↳ no of characters including null character

Eg: Jenny
1 2 3 4

print ⇒ Jenn 10

4 + for null = 5

NOTE:

* When the file will get stop reading we have Conditions

In file ⇒ [Jenny khatri Lamba,]

Condition 1

Char str[50];

(Eg)- fgets(str, 4, fp);

n

⇒ include code lines for file open.

It will read & print upto n-1 character and one for null

Condition 2 In file ⇒ [Jenny khatri Lamba]

(Eg) Char str[50];
fgets(str, 45, fp);

n

include code lines for file open statement

according to this it should read n-1 character i.e. 44 character but we have only 18 characters +1 for null character to terminate and it will stop.

Condition 3:- In file ⇒ [Jenny khatri Lamba (\n)
Jenny's Lecture (\n)
is the best (\n)]

* Buffer size should be above this lines of character

Case 1 * If file containing 3 lines; then it will read a file upto finding a new line character and it will stop reading and reads only first line.

(Eg) Char str[50];
fgets(str, 45, fp); ⇒ prints only first line

* So, how to read all the lines from a file; simply putting into while loop.

Case 2 Eg:-

```
char str[50];
while(!feof(fp))
{
    fgets(str, 10, fp);
    printf("%s", str);
}
```

Include code - lines for file open

(10-1) first 9 characters and 1 for null is read from file and print and next 9 character till goes on eof (end of file) the loop will continue & print the character (ie) the whole or all lines in a file

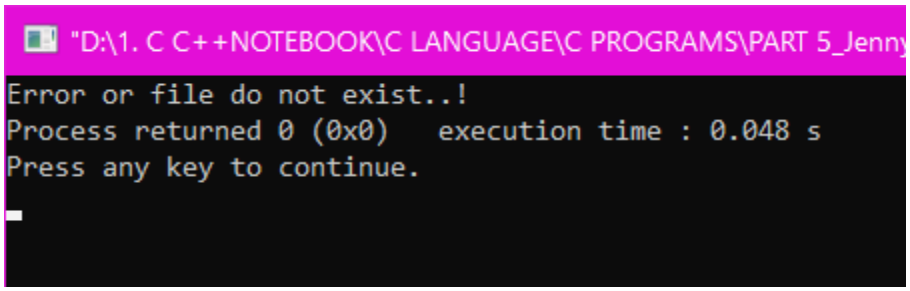
* here new line is not end of file and so it include new line also.

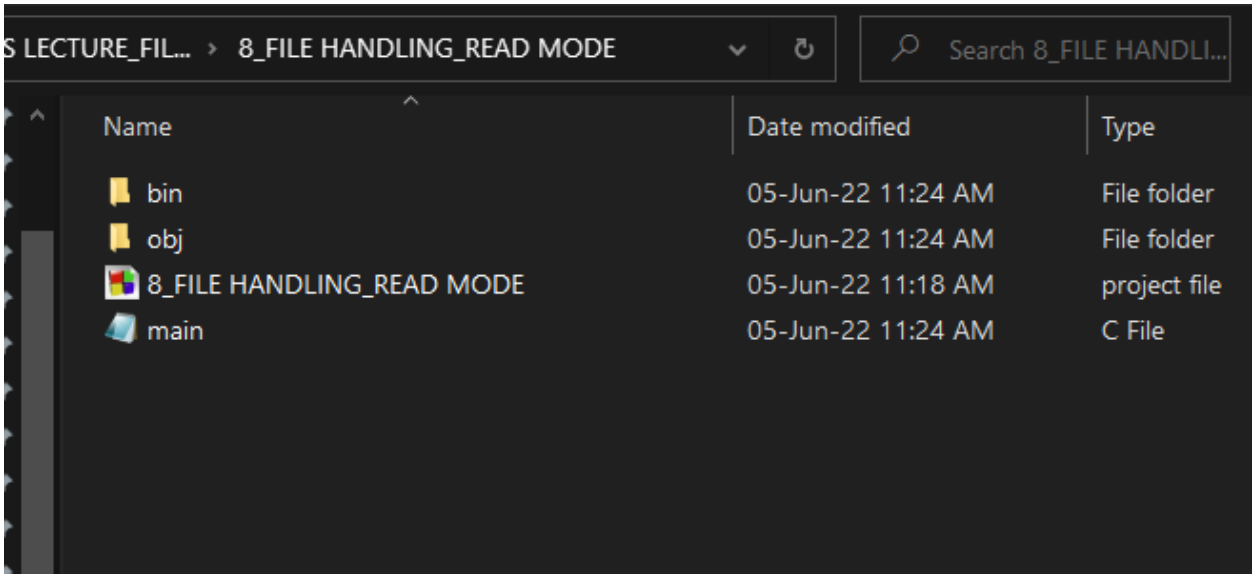
NOTE

* fScanf is for reading a file using formatted input function for file reading.

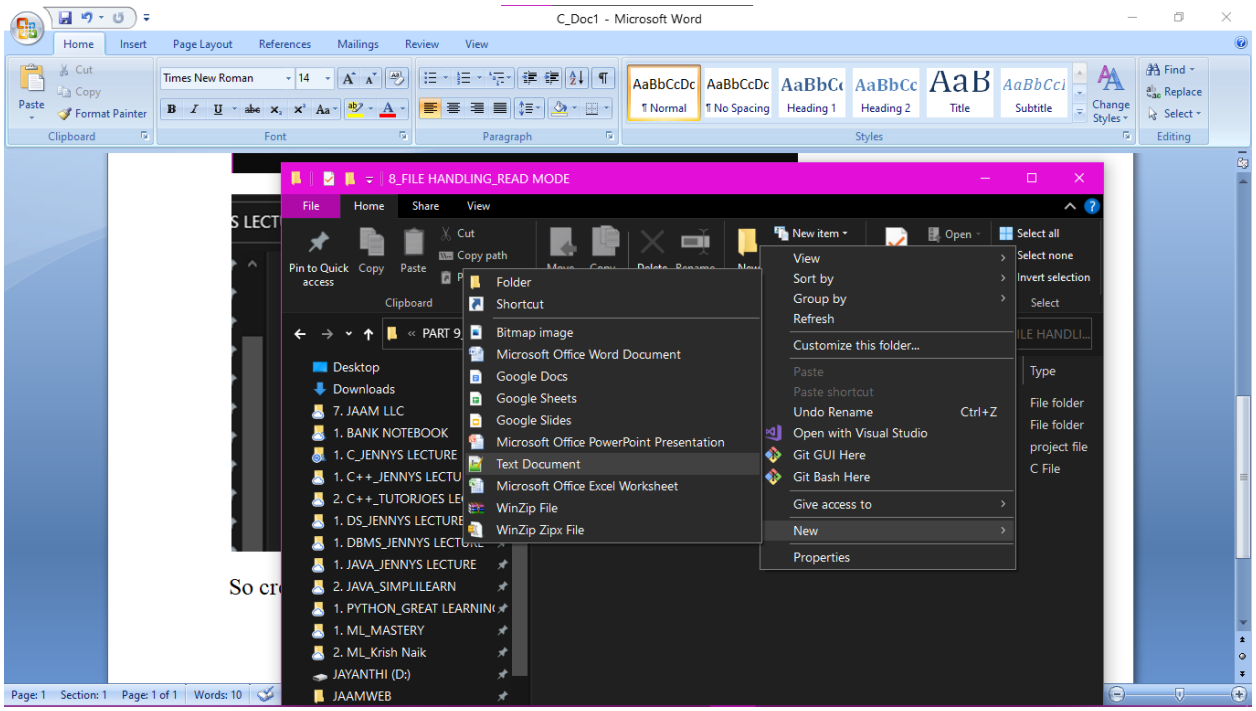
* This will be discussed in other lecture videos.

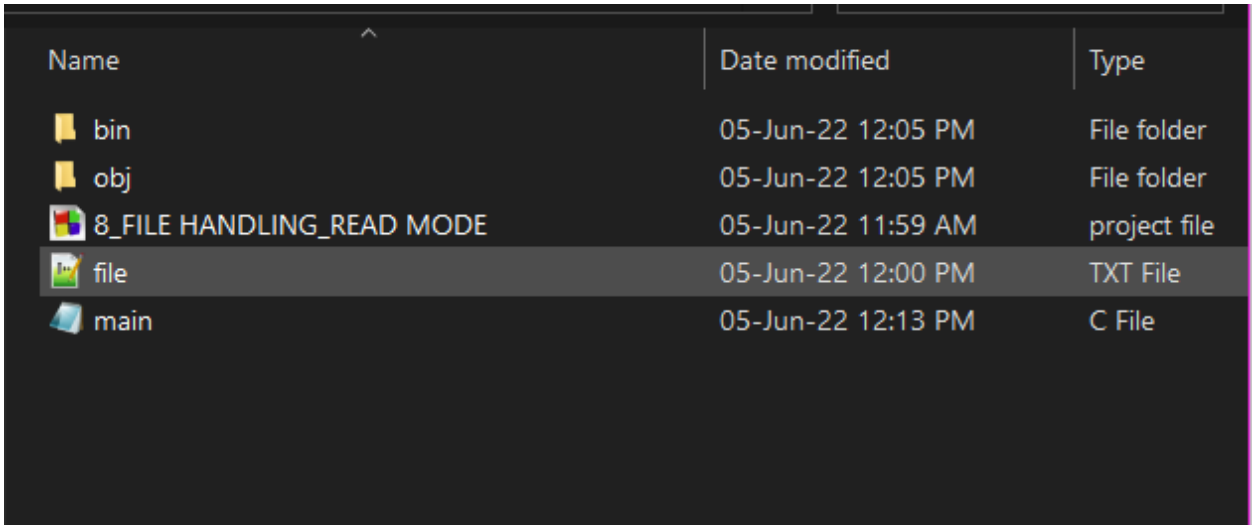
```c
#include <stdio.h>
#include <stdlib.h>
/** 8-FILE HANDLING-READ MODE **/
/** READING A CHARACTER FROM A FILE BY PRINTING IT IN OUTPUT SCREEN **/
int main()
{
    FILE *fp=NULL;
    fp=fopen("file.txt","r");//read mode will not create the file, so file should exist
    if(fp==NULL)
    {
     printf("Error or file do not exist..!");
    }

    char ch;
    ch=fgetc(fp);

    fclose(fp);

    return 0;
}

```



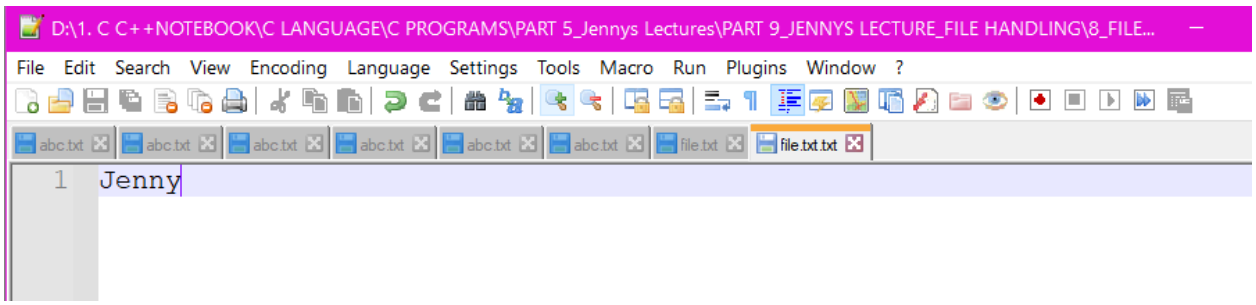


So create a file and read character from the file



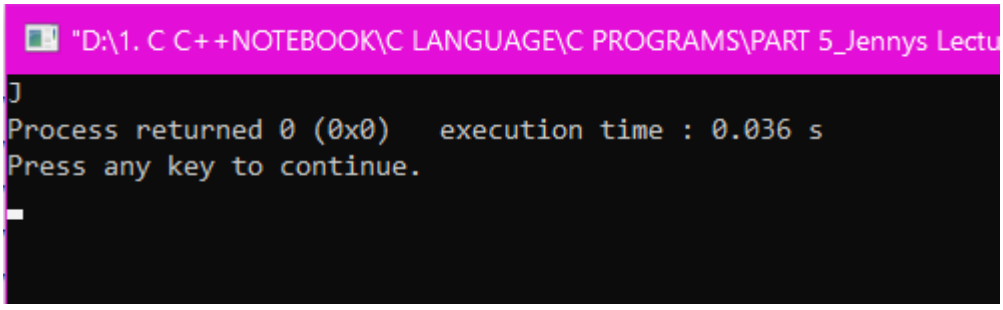


New file named “file.txt” is created and read a character from this file.

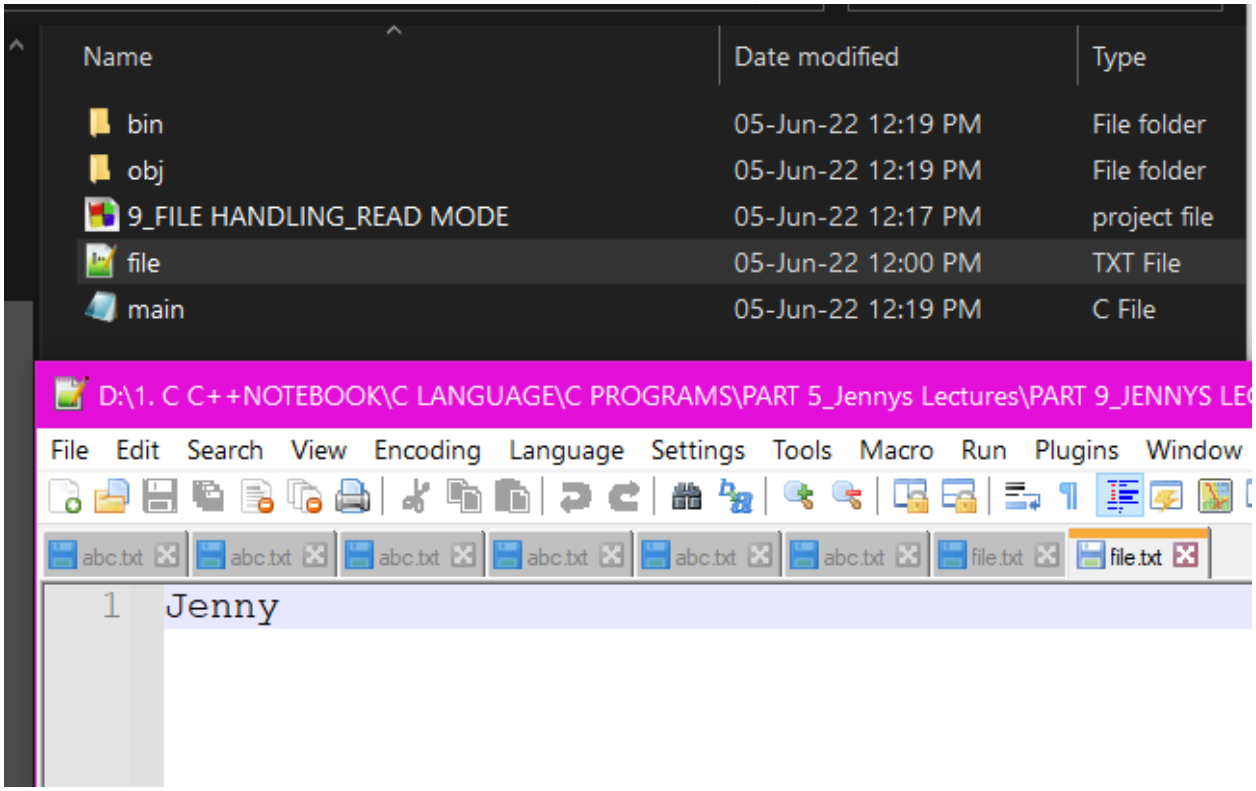


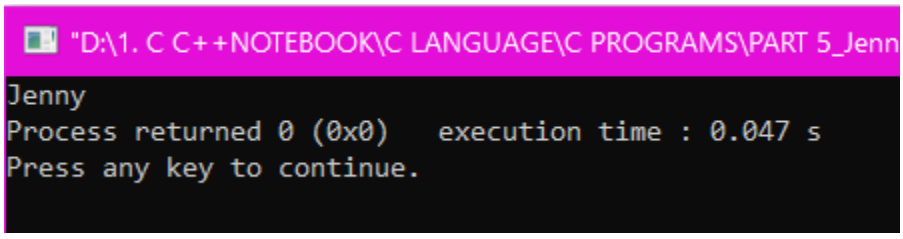
Content present in the file is given above.

```c
#include <stdio.h>
#include <stdlib.h>
/** 8-FILE HANDLING-READ MODE **/
/** READ A CHARACTER FROM FILE BY PRINTING IN OUTPUT SCREEN **/
int main()
{
    FILE *fp=NULL;
    char ch;
    fp=fopen("file.txt","r");
    if(fp==NULL)
    {
     printf("File do not exits/Error...!");
     exit(1);
    }
    ch=fgetc(fp);
    printf("%c",ch);

    fclose(fp);

    return 0;
}
```

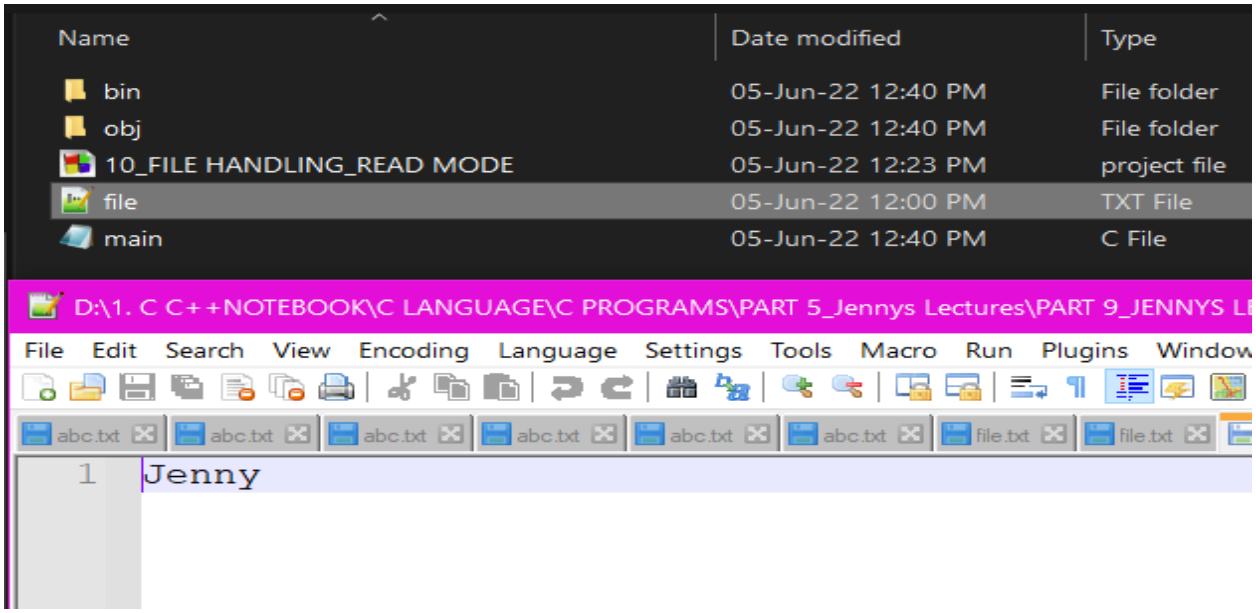


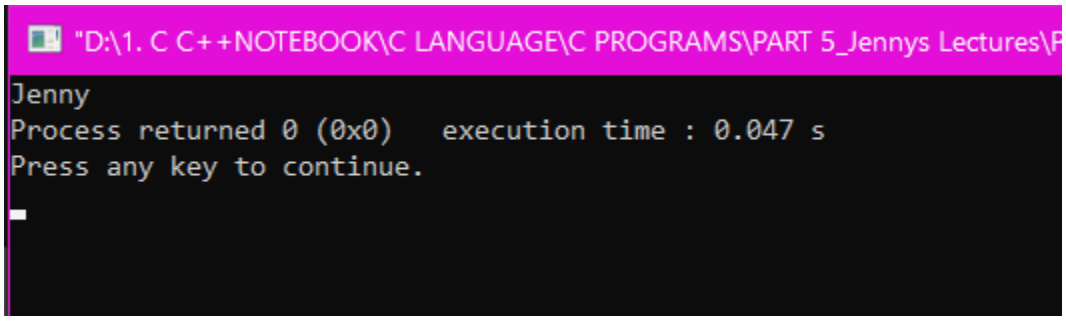
```c
#include <stdio.h>
#include <stdlib.h>
/** 9-FILE HANDLING-READ MODE **/
/** READ A ENTIRE STRING FROM FILE(CHARACTER BY CHARACTER) BY PRINTING IN OUTPUT SCREEN *
int main()
{
    FILE *fp=NULL;
    char ch;
    fp=fopen("file.txt","r");
    if(fp==NULL)
    {
     printf("File do not exits/Error...!");
     exit(1);
    }
    while(!feof(fp))
    {
    ch=fgetc(fp);
    printf("%c",ch);
    }
    fclose(fp);

    return 0;
}
```



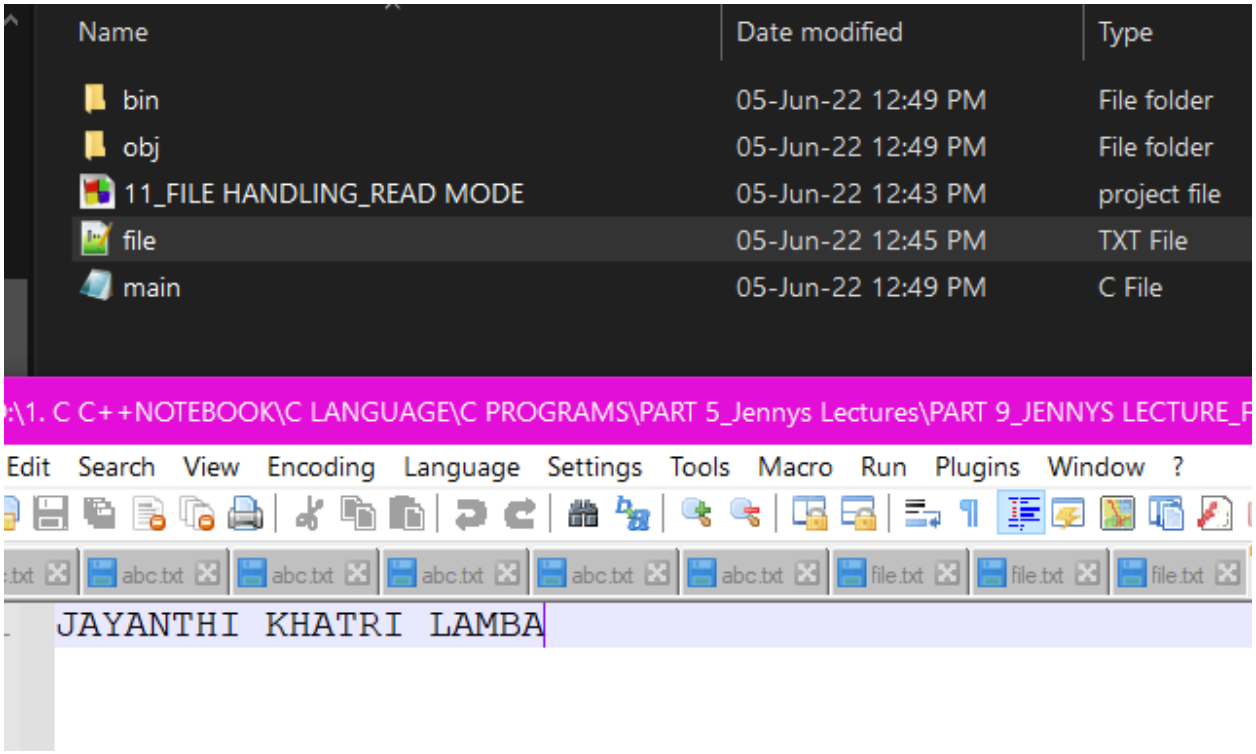


```c
/** 10-FILE HANDLING-READ MODE **/
/** READING ENTIRE STRING FROM FILE USING fgets() function **/
int main()
{
    FILE *fp=NULL;
    char str[6];//Buffer should have number of character and one null character
   //In our file we have Jenny containing 5 character and to store this in buffer
   // we need 5+1=6 that is the size of array buffer


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

   if(fp==NULL)
   {
    printf("Error/File do not exist..!");
    exit(1);
   }

   fgets(str,6,fp);
   //3 arguments: name of stirng, no of characters to read including null character
   // that needed to be stored inside the buffer and file pointer
   printf("%s",str);

   return 0;
}
```



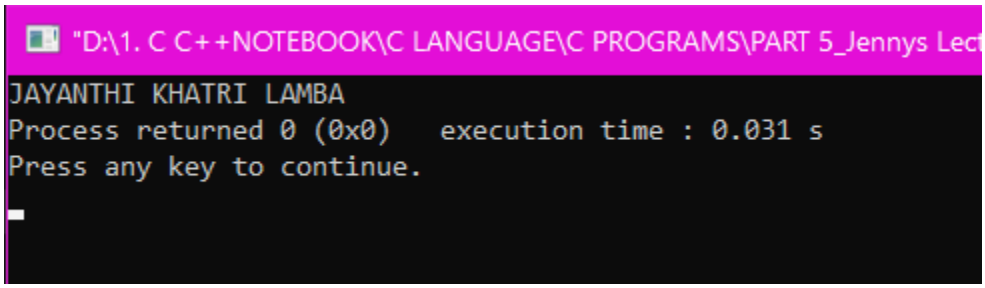


```c
#include <stdio.h>
#include <stdlib.h>
/** 11-FILE HANDLING-READ MODE **/
/** READING ENTIRE STRING OF SENTENCE FROM FILE USING fgets() function **/
int main()
{
    FILE *fp=NULL;
    char str[50];
    fp=fopen("file.txt","r");
    if(fp==NULL)
    {
     printf("File do not exist..!");
     exit(1);
    }
    fgets(str,45,fp);
    printf("%s",str);

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

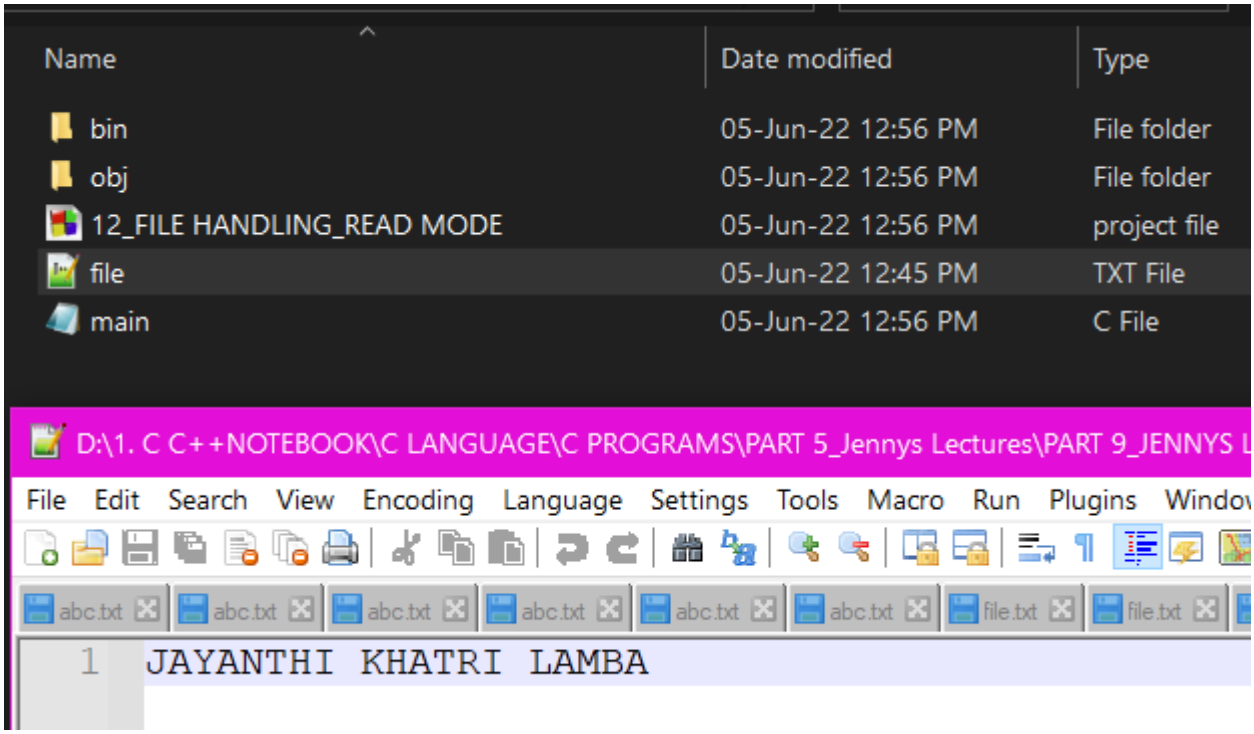


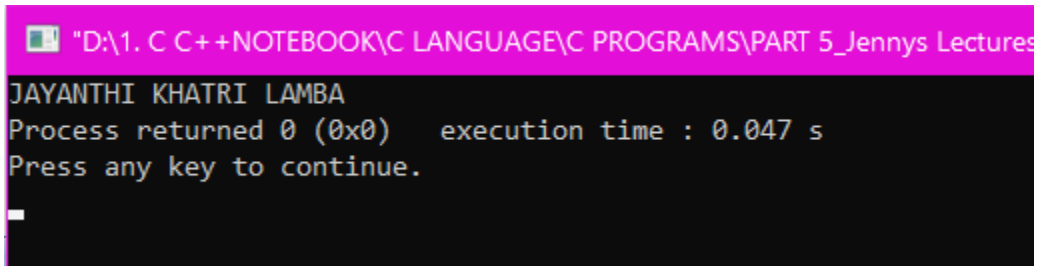
In our file we have 21 characters and one extra for null character, so size of buffer should be exactly 22 (i.e.) str[22], but we declared buffer size as str[50] and characters to read 45 even though we don’t have 45 characters, fgets() will read

upto the 21 characters and put null characters at end since there is no other character in our file.

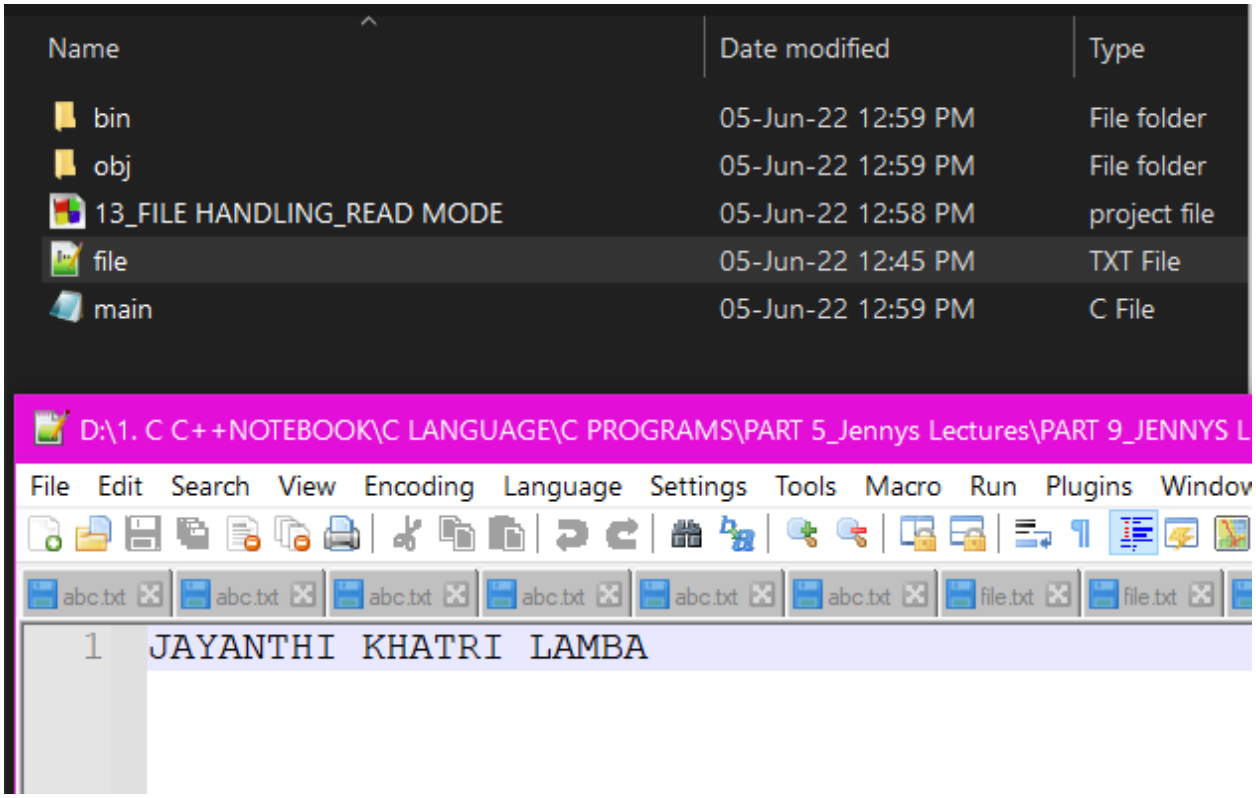


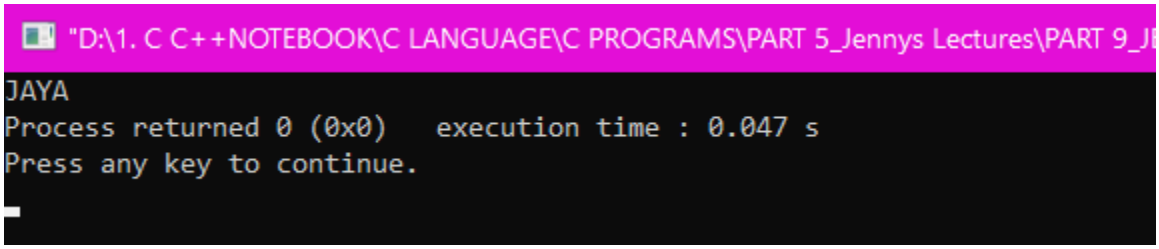

```c
#include <stdio.h>
#include <stdlib.h>
/** 12-FILE HANDLING-READ MODE **/
/** READING ENTIRE STRING OF SENTENCE FROM FILE USING fgets() function **/
int main()
{
    FILE *fp=NULL;
    char str[22];
    fp=fopen("file.txt","r");
    if(fp==NULL)
    {
     printf("File do not exist..!");
     exit(1);
    }
    fgets(str,22,fp);
    printf("%s",str);

    fclose(fp);
    return 0;
}

```



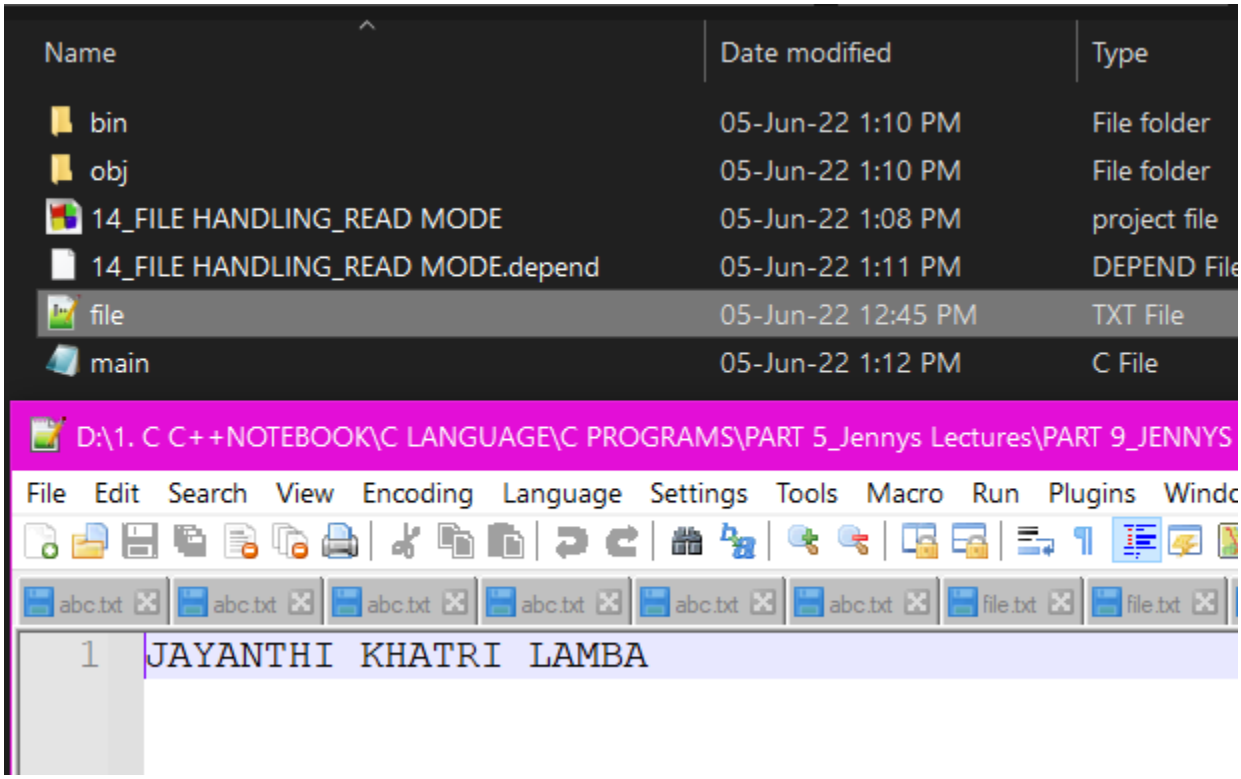


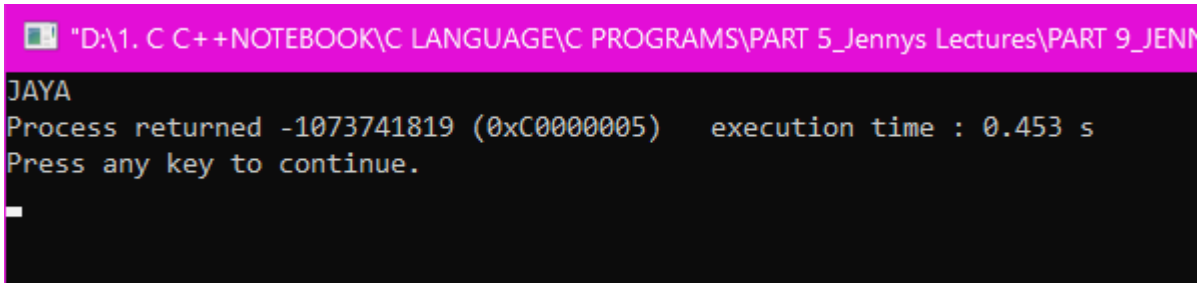

```c
#include <stdio.h>
#include <stdlib.h>
/** 13-FILE HANDLING-READ MODE **/
/** READING ENTIRE STRING OF SENTENCE FROM FILE USING fgets() function **/
int main()
{
    FILE *fp=NULL;
    char str[22];
    fp=fopen("file.txt","r");
    if(fp==NULL)
    {
     printf("File do not exist..!");
     exit(1);
    }
    fgets(str,5,fp);
    printf("%s",str);

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





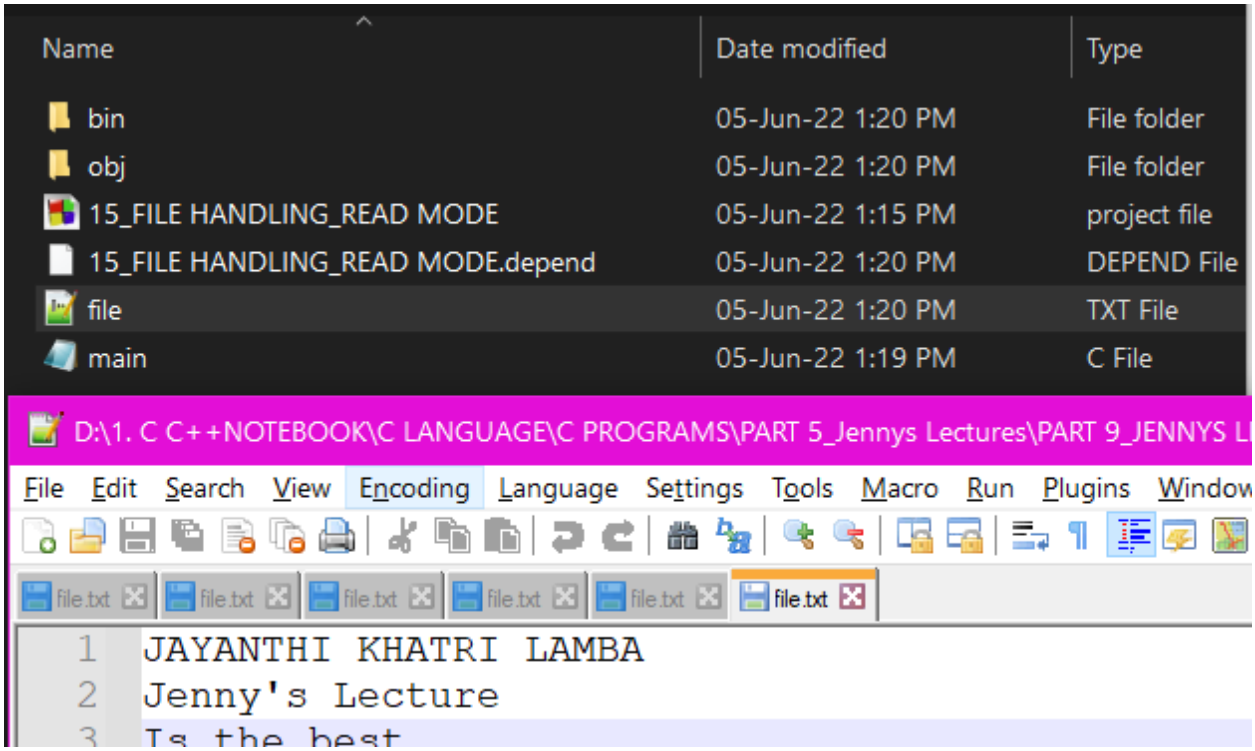

```c
#include <stdio.h>
#include <stdlib.h>
/** 14-FILE HANDLING-READ MODE **/
/** READING ENTIRE STRING OF SENTENCE FROM FILE USING fgets() function **/
int main()
{
    FILE *fp=NULL;
    char str[2];
    fp=fopen("file.txt","r");
    if(fp==NULL)
    {
     printf("File do not exist..!");
     exit(1);
    }
    fgets(str,5,fp);
    //should not exceed buffer size str[2], we gave 5, it will read but occupies
    // memory of some other file using it or it is overwritten and cause problem
    printf("%s",str);

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

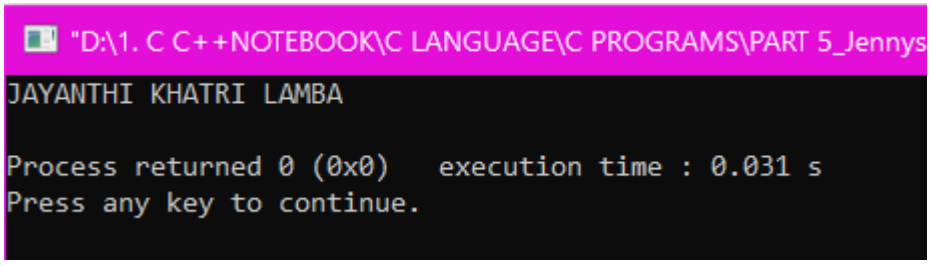




Process not returned with 0, something has happened and gives like warning, because we have overwritten the memory after the maximum size of the buffer.

```c
#include <stdio.h>
#include <stdlib.h>
/** 15-FILE HANDLING-READ MODE **/
/** READING MORE THAN ONE SENTENCE FROM FILE USING fgets() function **/
int main()
{
    FILE *fp=NULL;
    char str[50];
    fp=fopen("file.txt","r");
    if(fp==NULL)
    {
     printf("File do not exist..!");
     exit(1);
    }
    fgets(str,50,fp);
    //fgets() will consider the new line character as end of line and stops reading
    // a file and terminates with null character
    printf("%s",str);
    //So only one sentence is only read from a file

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





```c
#include <stdio.h>
#include <stdlib.h>
/** 16-FILE HANDLING-READ MODE **/
/** READING MORE THAN ONE SENTENCE FROM FILE USING fgets() function **/
int main()
{
    FILE *fp=NULL;
    char str[7];
    fp=fopen("file.txt","r");
    if(fp==NULL)
    {
     printf("File do not exist..!");
     exit(1);
    }
    while(!feof(fp)) //buffer is overwritten every time, printed with new line till end
    {
    fgets(str,7,fp);//read first 6 character and put in buffer with one null & prints it
    printf("%s",str);
    }
    fclose(fp);
    return 0;
}
```