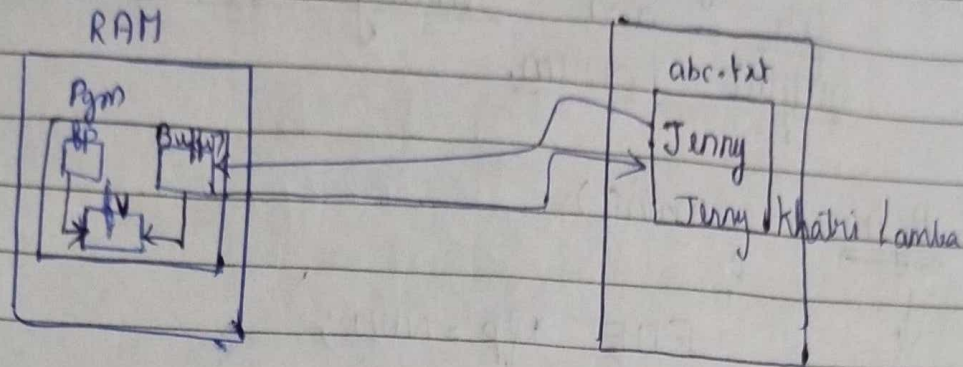


Date _____
Page _____

C_122 \Rightarrow File Handling in C \Rightarrow Part 4

Read from file in C



* whenever we read the content of a file; the copy of content of 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,

- \rightarrow `fgetc` (read a character)
- \rightarrow `fgets` (read a string)

NOTE:

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

\rightarrow So write a code 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("i.e.", ch);
    fclose(fp);
}
```

O/p : It will print 'I' 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:-

fgetc is predefined function

Date _____
Page _____

char ch;

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

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

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

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

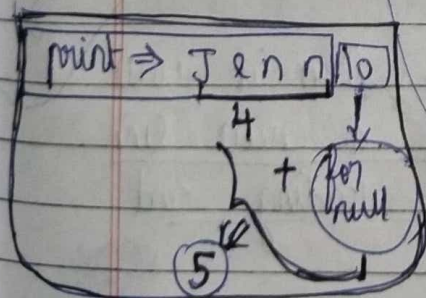
main()

{

NOTE

→ no. of
characters
including
null character

Eg: Jenny
1 2 3 4



FILE *fp = NULL;

char str[10];

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

if (fp == NULL)

{
 printf("error or file ^{not} exist");
 exit(1);
}

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

max 9 in this case if we give 10

fgets(str, 5, fp); ⇒ 3 arguments

name of
string

no. of
character

file pointer

printf("%s", str);

fclose(fp);

NOTE

* when the file will get stop
reading we have Conditions

In file \Rightarrow Jenny khatri Lamba

Condition 1

Char str[50];

(Eg) fgetc (str, 4, fp);

\downarrow
n

\Rightarrow include code
lines
for file
open.

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

Condition 2

In file \Rightarrow Jenny khatri Lamba

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

\downarrow
n

include code lines
for file
open in between

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.

* Buffer size
should be
above
this
lines of
characters
50

Condition 3:-

In file \Rightarrow

Jenny khatri Lamba (n)
Jenny's Lecture (n)
is the best (n)

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];

fgetc (str, 45, fp);

\Rightarrow prints only first
line

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

Case 2

eg:-

char str[100];

while(!feof(fp))

{

fgets(str, 100, 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 ~~for~~ function for file reading.

* This will be discussed in other lecture
videos.


```

1  #include <stdio.h>
2  #include <stdlib.h>
3  /** 8-FILE HANDLING-READ MODE **/
4  /** READING A CHARACTER FROM A FILE BY PRINTING IT IN OUTPUT SCREEN **/
5  int main()
6  {
7      FILE *fp=NULL;
8      fp=fopen("file.txt", "r"); //read mode will not create the file, so file should exist
9      if (fp==NULL)
10     {
11         printf("Error or file do not exist..!");
12     }
13
14     char ch;
15     ch=fgetc(fp);
16
17     fclose(fp);
18
19     return 0;
20 }
21

```

"D:\1. C C++NOTEBOOK\C LANGUAGE\C PROGRAMS\PART 5_Jenny

```

Error or file do not exist..!
Process returned 0 (0x0)   execution time : 0.048 s
Press any key to continue.

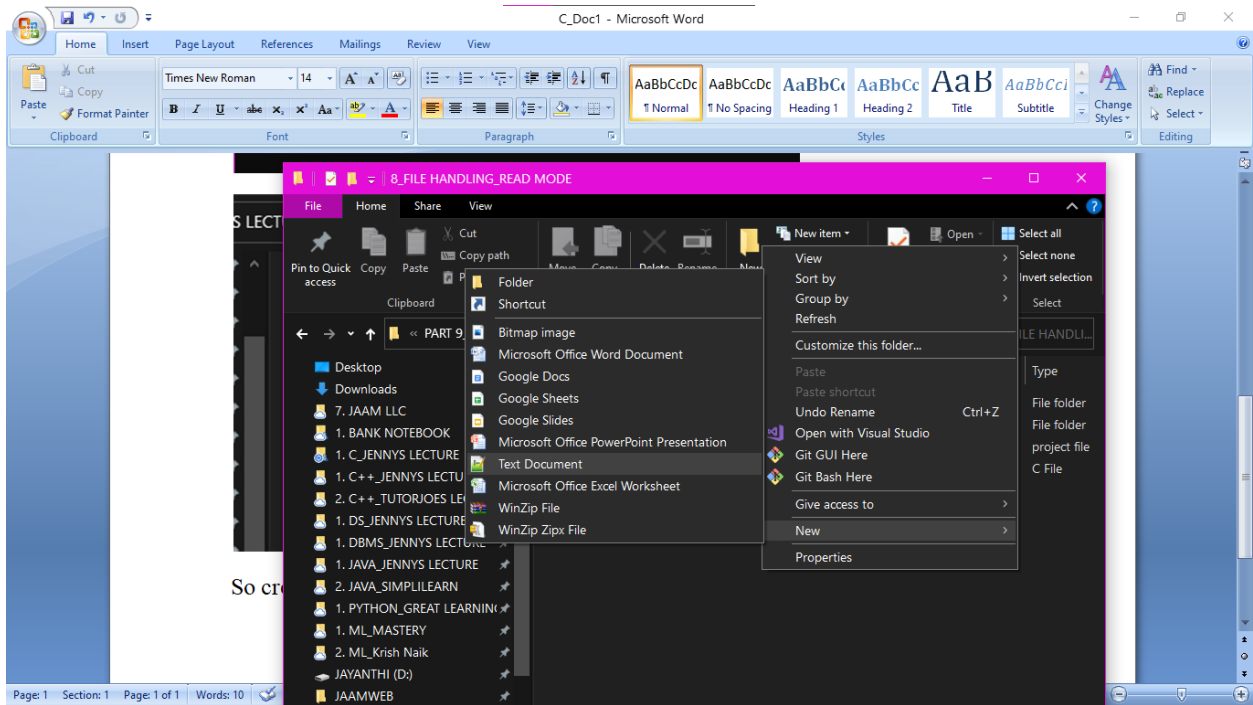
```

S LECTURE_FIL... > 8_FILE HANDLING_READ MODE

Search 8_FILE HANDLI...

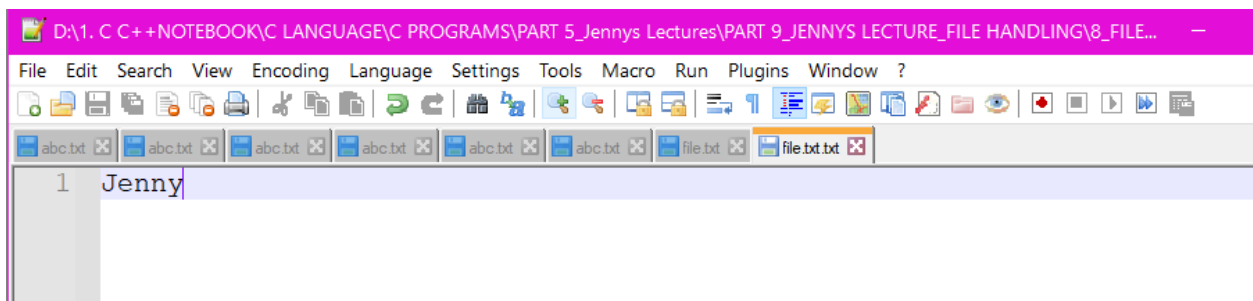
Name	Date modified	Type
bin	05-Jun-22 11:24 AM	File folder
obj	05-Jun-22 11:24 AM	File folder
8_FILE HANDLING_READ MODE	05-Jun-22 11:18 AM	project file
main	05-Jun-22 11:24 AM	C File

So create a file and read character from the file



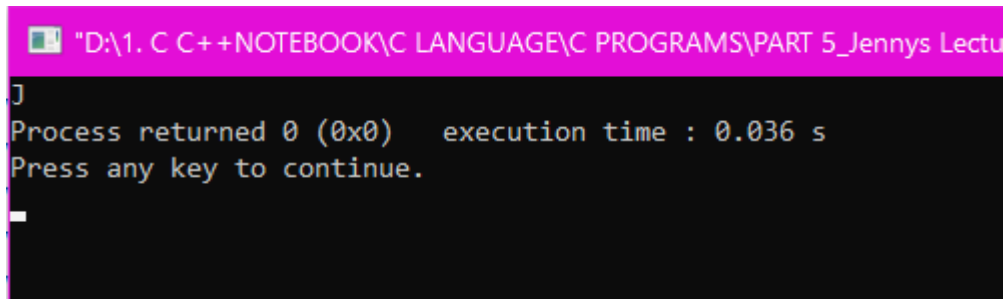
Name	Date modified	Type
bin	05-Jun-22 12:05 PM	File folder
obj	05-Jun-22 12:05 PM	File folder
8_FILE HANDLING_READ MODE	05-Jun-22 11:59 AM	project file
file	05-Jun-22 12:00 PM	TXT File
main	05-Jun-22 12:13 PM	C File

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



Content present in the file is given above.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  /** 8-FILE HANDLING-READ MODE **/
4  /** READ A CHARACTER FROM FILE BY PRINTING IN OUTPUT SCREEN **/
5  int main()
6  {
7      FILE *fp=NULL;
8      char ch;
9      fp=fopen("file.txt", "r");
10     if(fp==NULL)
11     {
12         printf("File do not exists/Error...!");
13         exit(1);
14     }
15     ch=fgetc(fp);
16     printf("%c", ch);
17
18     fclose(fp);
19
20     return 0;
21 }
22
```



"D:\1. C C++NOTEBOOK\C LANGUAGE\C PROGRAMS\PART 5_Jennys Lectu

J

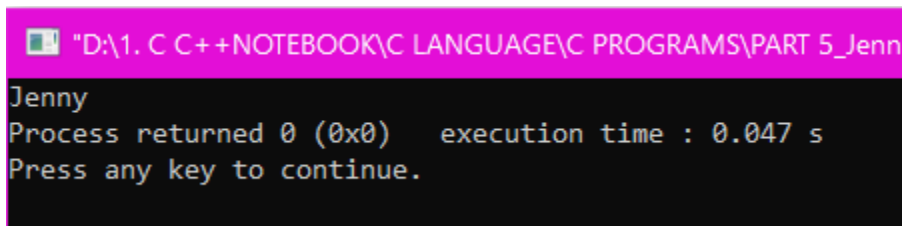
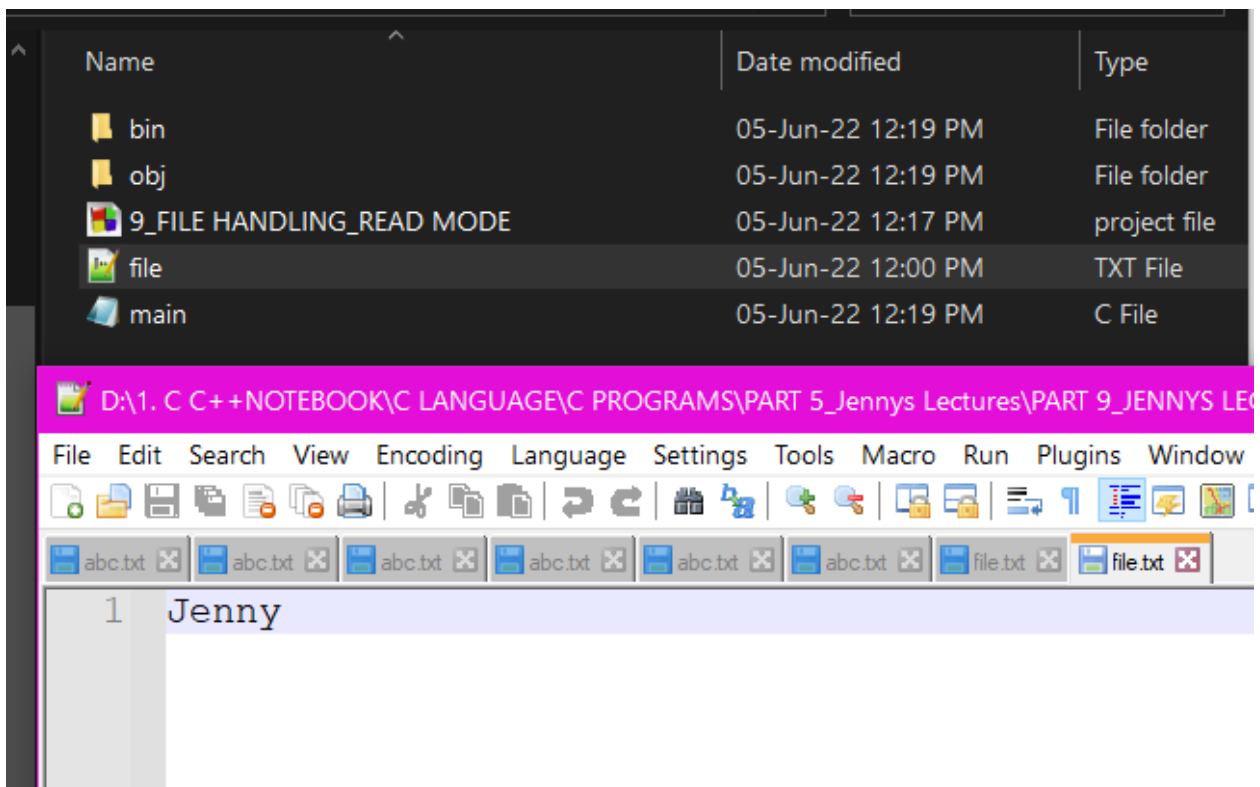
Process returned 0 (0x0) execution time : 0.036 s

Press any key to continue.


```

1  #include <stdio.h>
2  #include <stdlib.h>
3  /** 9-FILE HANDLING-READ MODE **/
4  /** READ A ENTIRE STRING FROM FILE (CHARACTER BY CHARACTER) BY PRINTING IN OUTPUT SCREEN **/
5  int main()
6  {
7      FILE *fp=NULL;
8      char ch;
9      fp=fopen("file.txt","r");
10     if (fp==NULL)
11     {
12         printf("File do not exists/Error...!");
13         exit(1);
14     }
15     while (!feof(fp))
16     {
17         ch=fgetc(fp);
18         printf("%c",ch);
19     }
20     fclose(fp);
21
22     return 0;
23 }
24

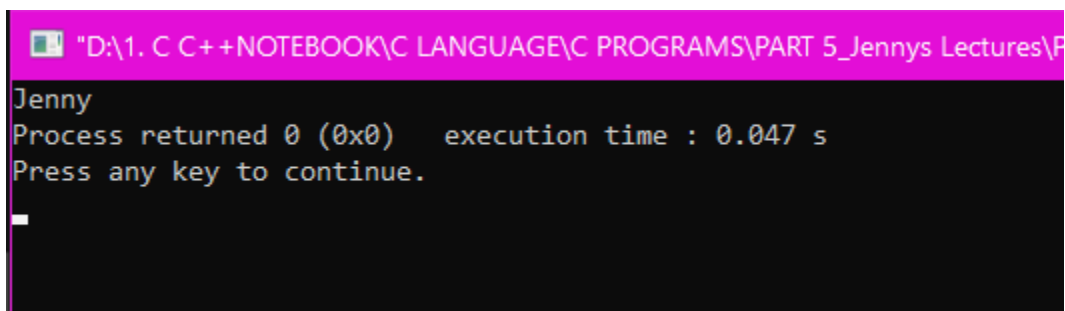
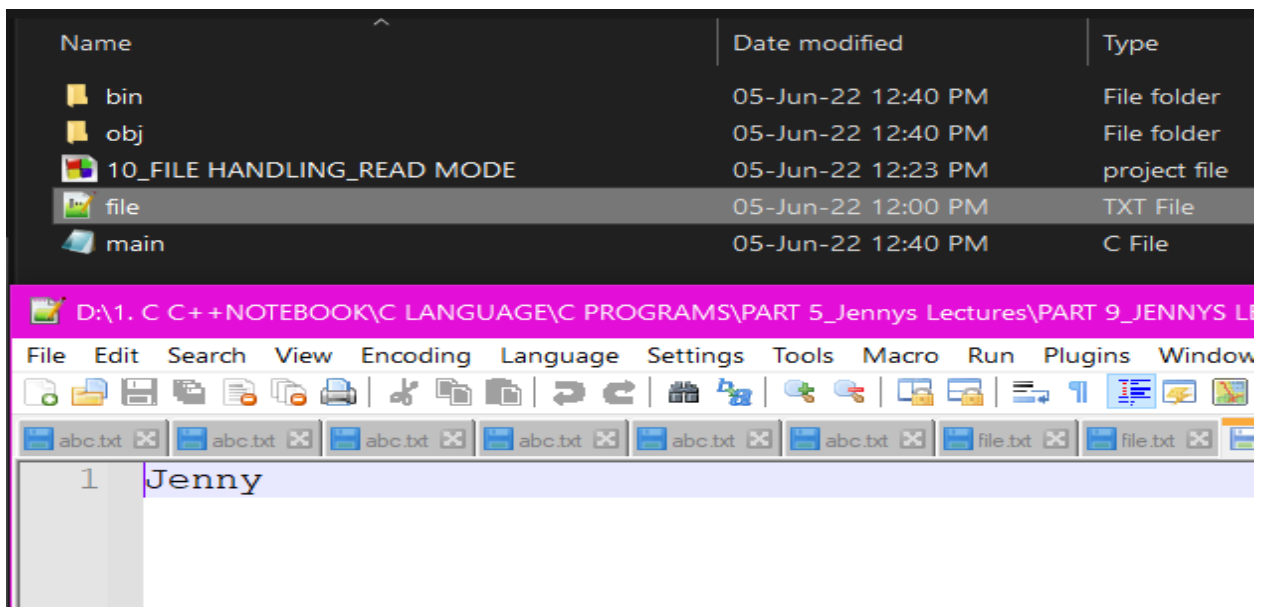
```



```

3  /** 10-FILE HANDLING-READ MODE **/
4  /** READING ENTIRE STRING FROM FILE USING fgets() function **/
5  int main()
6  {
7      FILE *fp=NULL;
8      char str[6]; //Buffer should have number of character and one null character
9      //In our file we have Jenny containing 5 character and to store this in buffer
10     // we need 5+1=6 that is the size of array buffer
11
12
13     fp=fopen("file.txt", "r");
14
15     if(fp==NULL)
16     {
17         printf("Error/File do not exist..!");
18         exit(1);
19     }
20
21     fgets(str, 6, fp);
22     //3 arguments: name of string, no of characters to read including null character
23     // that needed to be stored inside the buffer and file pointer
24     printf("%s", str);
25
26     return 0;
27 }
28

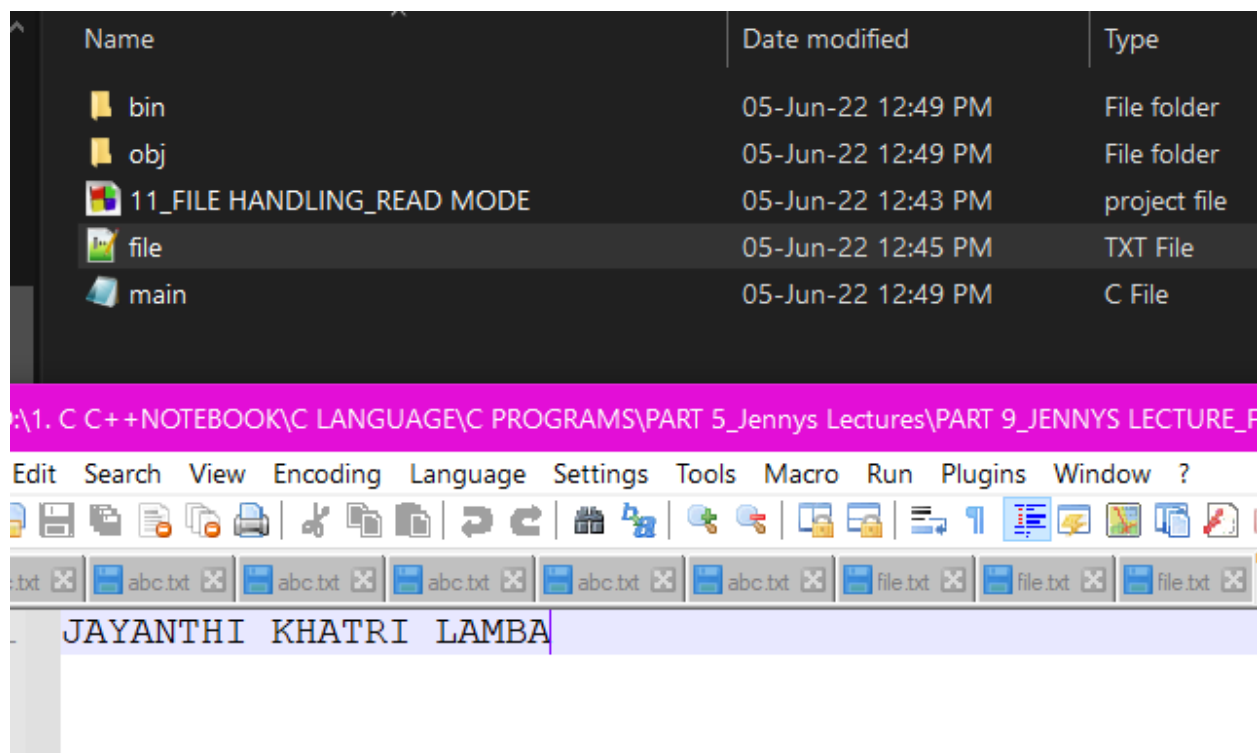
```




```

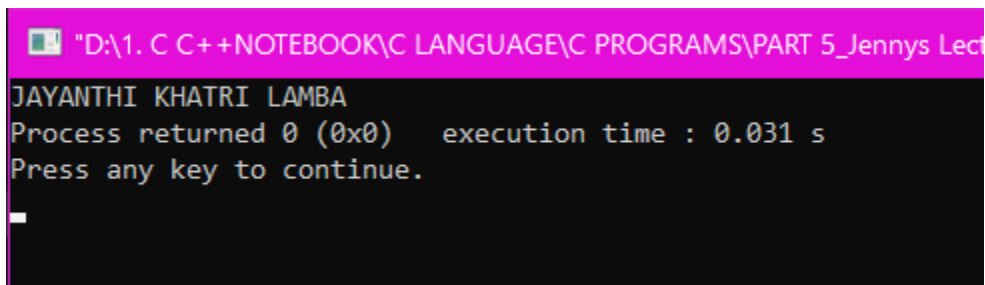
1  #include <stdio.h>
2  #include <stdlib.h>
3  /** 11-FILE HANDLING-READ MODE **/
4  /** READING ENTIRE STRING OF SENTENCE FROM FILE USING fgets() function **/
5  int main()
6  {
7      FILE *fp=NULL;
8      char str[50];
9      fp=fopen("file.txt", "r");
10     if(fp==NULL)
11     {
12         printf("File do not exist..!");
13         exit(1);
14     }
15     fgets(str, 45, fp);
16     printf("%s", str);
17
18     fclose(fp);
19     return 0;
20 }
21

```



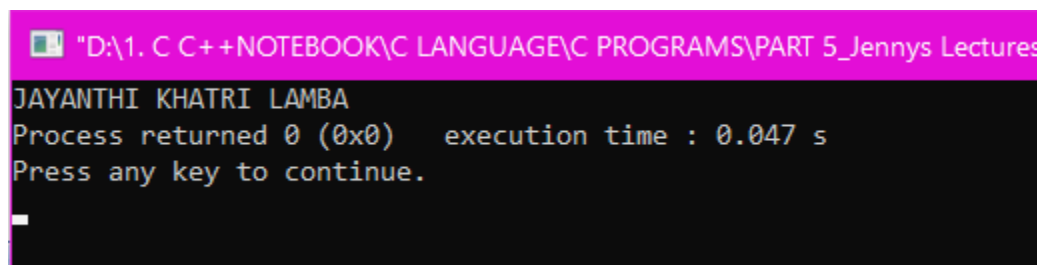
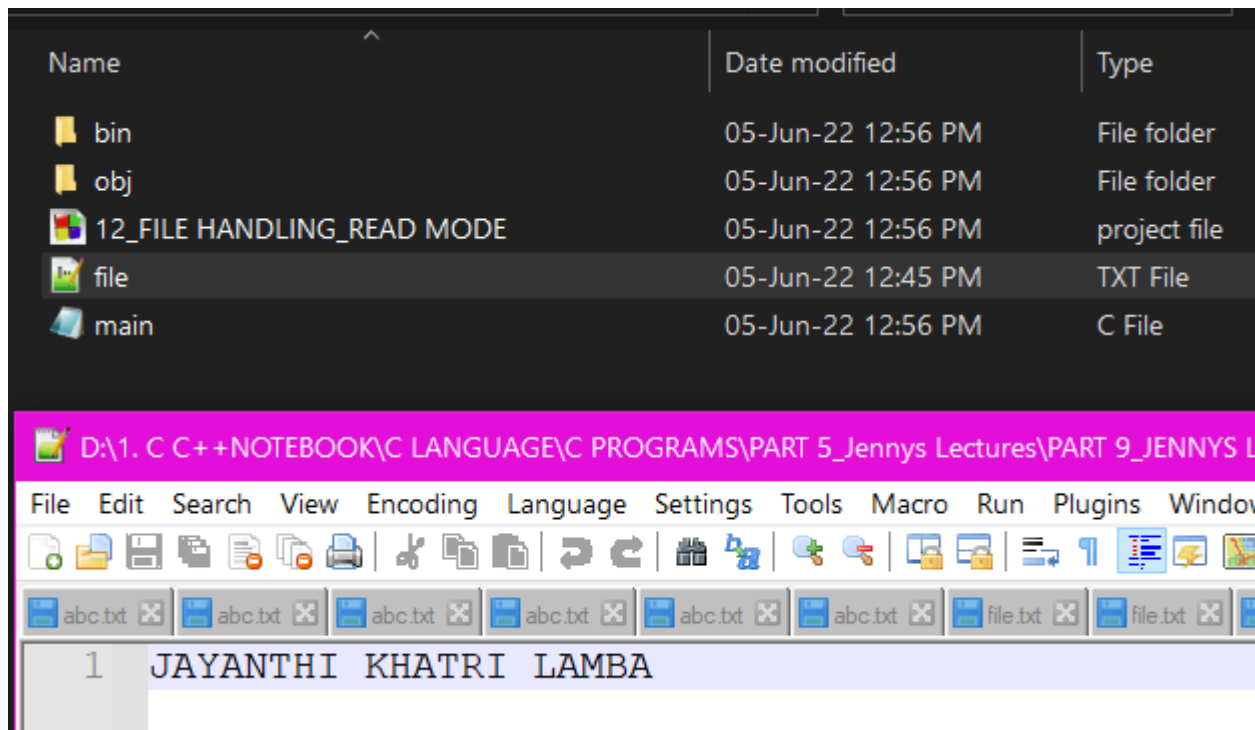
In our file we have 21 characters and one extra for null character, so size of buffer should be exactly 22 (i.e.) str[22], but we declared buffer size as str[50] and characters to read 45 even though we don't have 45 characters, fgets() will read

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

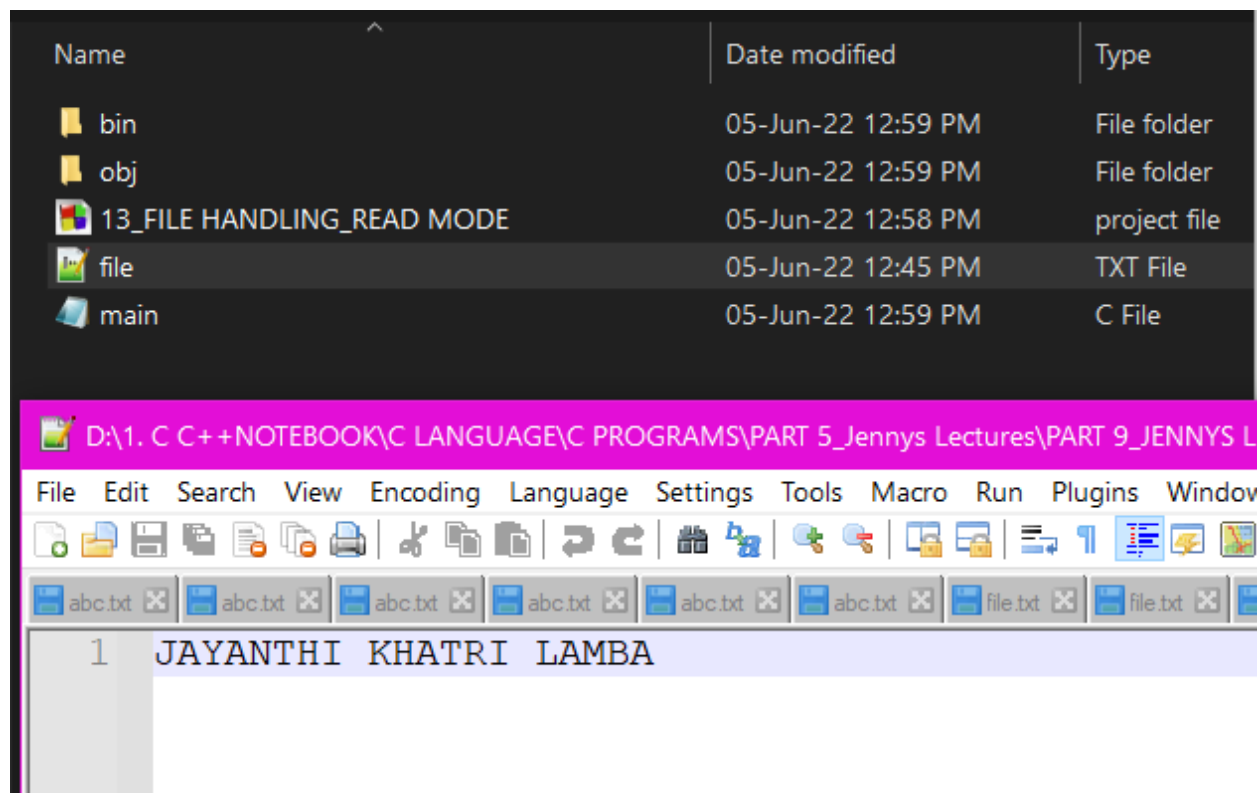


```
"D:\1. C C++ NOTEBOOK\C LANGUAGE\C PROGRAMS\PART 5_Jennys Lect
JAYANTHI KHATRI LAMBA
Process returned 0 (0x0)   execution time : 0.031 s
Press any key to continue.
_
```

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  /** 12-FILE HANDLING-READ MODE **/
4  /** READING ENTIRE STRING OF SENTENCE FROM FILE USING fgets() function **/
5  int main()
6  {
7      FILE *fp=NULL;
8      char str[22];
9      fp=fopen("file.txt", "r");
10     if(fp==NULL)
11     {
12         printf("File do not exist..!");
13         exit(1);
14     }
15     fgets(str, 22, fp);
16     printf("%s", str);
17
18     fclose(fp);
19     return 0;
20 }
21
```

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  /** 13-FILE HANDLING-READ MODE **/
4  /** READING ENTIRE STRING OF SENTENCE FROM FILE USING fgets() function **/
5  int main()
6  {
7      FILE *fp=NULL;
8      char str[22];
9      fp=fopen("file.txt","r");
10     if(fp==NULL)
11     {
12         printf("File do not exist..!");
13         exit(1);
14     }
15     fgets(str,5,fp);
16     printf("%s",str);
17
18     fclose(fp);
19     return 0;
20 }
21
```



"D:\1. C C++NOTEBOOK\C LANGUAGE\C PROGRAMS\PART 5_Jennys Lectures\PART 9_J

JAYA

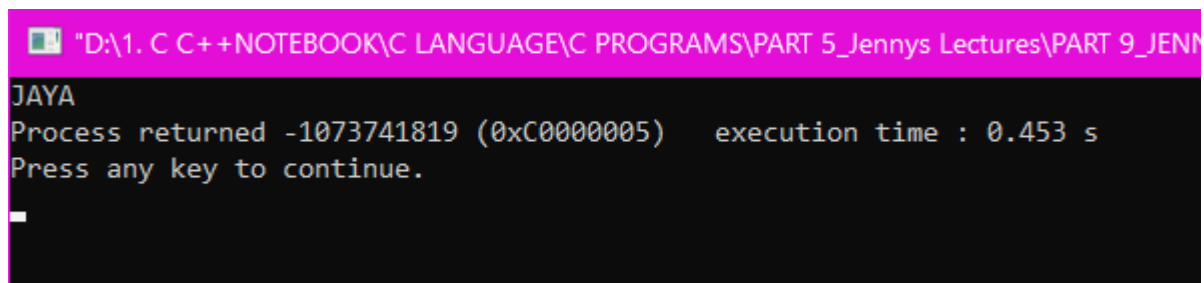
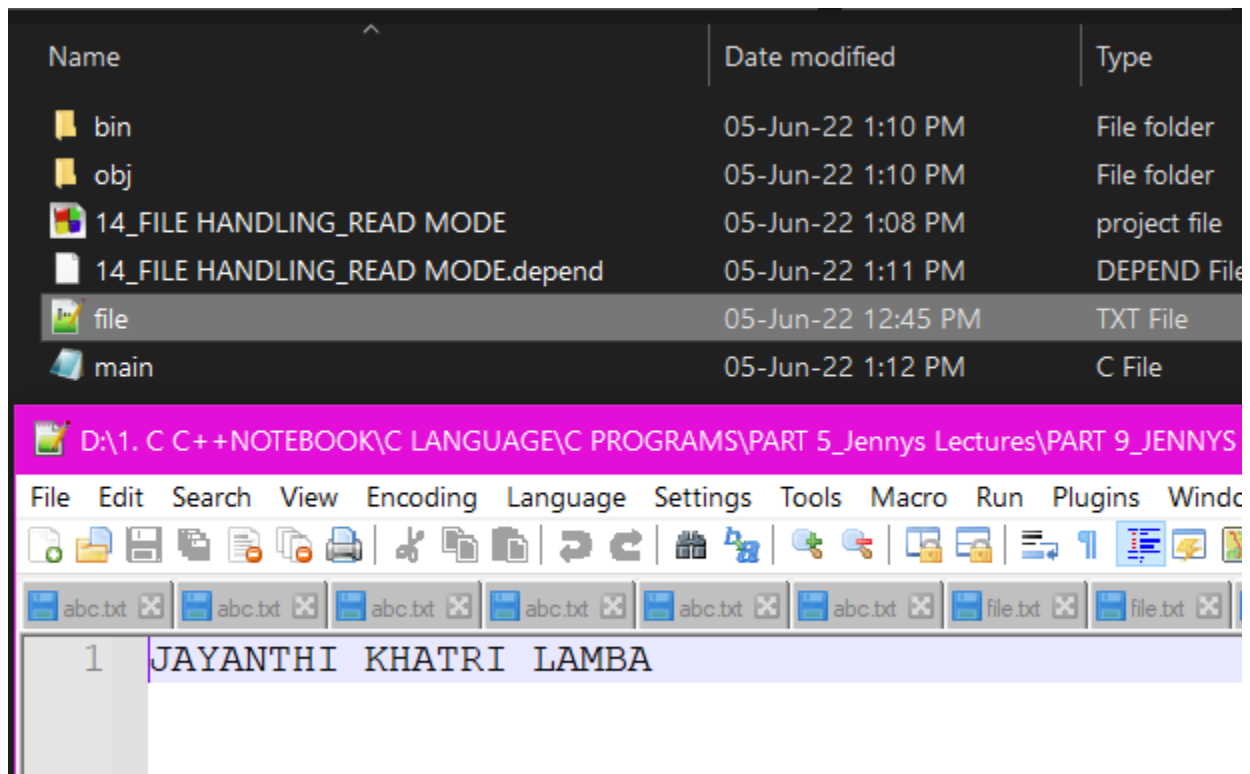
Process returned 0 (0x0) execution time : 0.047 s

Press any key to continue.

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  /** 14-FILE HANDLING-READ MODE **/
4  /** READING ENTIRE STRING OF SENTENCE FROM FILE USING fgets() function **/
5  int main()
6  {
7      FILE *fp=NULL;
8      char str[2];
9      fp=fopen("file.txt","r");
10     if(fp==NULL)
11     {
12         printf("File do not exist..!");
13         exit(1);
14     }
15     fgets(str,5,fp);
16     //should not exceed buffer size str[2], we gave 5, it will read but occupies
17     // memory of some other file using it or it is overwritten and cause problem
18     printf("%s",str);
19
20     fclose(fp);
21     return 0;
22 }
23

```

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.

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  /** 15-FILE HANDLING-READ MODE **/
4  /** READING MORE THAN ONE SENTENCE FROM FILE USING fgets() function **/
5  int main()
6  {
7      FILE *fp=NULL;
8      char str[50];
9      fp=fopen("file.txt","r");
10     if(fp==NULL)
11     {
12         printf("File do not exist..!");
13         exit(1);
14     }
15     fgets(str,50,fp);
16     //fgets() will consider the new line character as end of line and stops reading
17     // a file and terminates with null character
18     printf("%s",str);
19     //So only one sentence is only read from a file
20
21     fclose(fp);
22     return 0;
23 }
24

```

Name	Date modified	Type
bin	05-Jun-22 1:20 PM	File folder
obj	05-Jun-22 1:20 PM	File folder
15_FILE HANDLING_READ MODE	05-Jun-22 1:15 PM	project file
15_FILE HANDLING_READ MODE.depend	05-Jun-22 1:20 PM	DEPEND File
file	05-Jun-22 1:20 PM	TXT File
main	05-Jun-22 1:19 PM	C File

D:\1. C C++NOTEBOOK\C LANGUAGE\C PROGRAMS\PART 5_Jennys Lectures\PART 9_JENNYS L

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window

file.txt x file.txt x file.txt x file.txt x file.txt x file.txt x

```

1 JAYANTHI KHATRI LAMBA
2 Jenny's Lecture
3 Is the best

```

"D:\1. C C++NOTEBOOK\C LANGUAGE\C PROGRAMS\PART 5_Jennys

JAYANTHI KHATRI LAMBA

Process returned 0 (0x0) execution time : 0.031 s
Press any key to continue.

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  /** 16-FILE HANDLING-READ MODE **/
4  /** READING MORE THAN ONE SENTENCE FROM FILE USING fgets() function **/
5  int main()
6  {
7      FILE *fp=NULL;
8      char str[7];
9      fp=fopen("file.txt","r");
10     if(fp==NULL)
11     {
12         printf("File do not exist..!");
13         exit(1);
14     }
15     while(!feof(fp)) //buffer is overwritten every time, printed with new line till end
16     {
17         fgets(str,7,fp); //read first 6 character and put in buffer with one null & prints it
18         printf("%s",str);
19     }
20     fclose(fp);
21     return 0;
22 }

```

Name	Date modified	Type
bin	05-Jun-22 1:48 PM	File folder
obj	05-Jun-22 1:48 PM	File folder
16_FILE HANDLING_READ MODE	05-Jun-22 1:41 PM	project file
file	05-Jun-22 1:20 PM	TXT File
main	05-Jun-22 1:50 PM	C File

D:\1. C C++NOTEBOOK\C LANGUAGE\C PROGRAMS\PART 5_Jennys Lectures\PART 9_JENNYS

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window

File.txt File.txt File.txt File.txt File.txt File.txt

```

1 JAYANTHI KHATRI LAMBA
2 Jenny's Lecture
3 Is the best

```

"D:\1. C C++NOTEBOOK\C LANGUAGE\C PROGRAMS\PART 5_Jennys L

```

JAYANTHI KHATRI LAMBA
Jenny's Lecture
Is the best
Process returned 0 (0x0)   execution time : 0.047 s
Press any key to continue.

```