Python File Handling

A file operation can be done in following order

* Open a file
* Read or write – Performing operation
* Close the file

Opening a file –

File opening is done with open() function in Python. This function will accepts two arguments, file name and access mode in which the file is accessed.

Syntax –

File object =open(<file-name>,<access-mode>,<buffering>)

Access mode – some are below

r – means only read

rb – means only read in binary format

r+ --- means read and write

rb+ --- means read and write both in binary format

w --- means only write

wb – write only binary format

w+ --- means write and read both

wb+ ----- write and read both in binary format

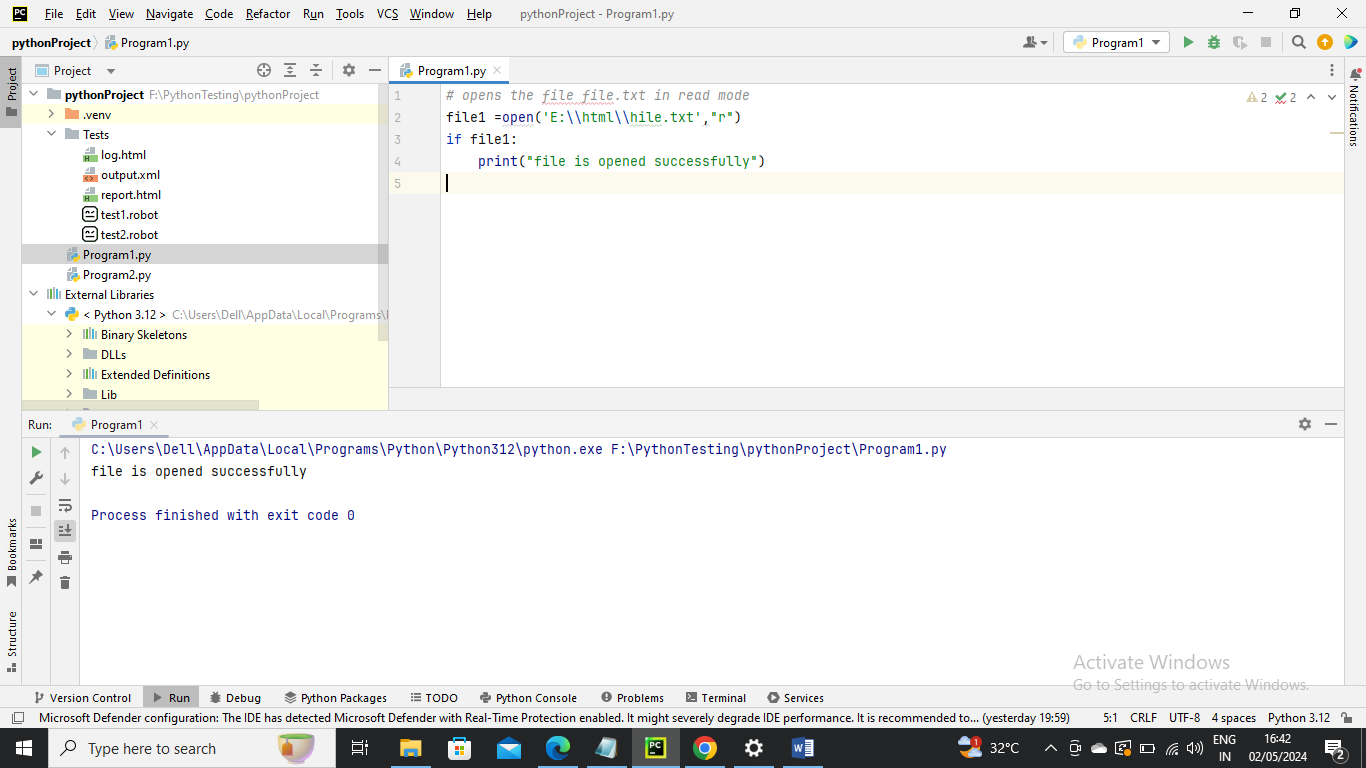
a ---- opens file in append mode

ab --- opens the file in append mode in binary format

a+ ---- opens file in append and read mode

ab+ ----- opens the file in append and read mode in binary format

Program for read mode

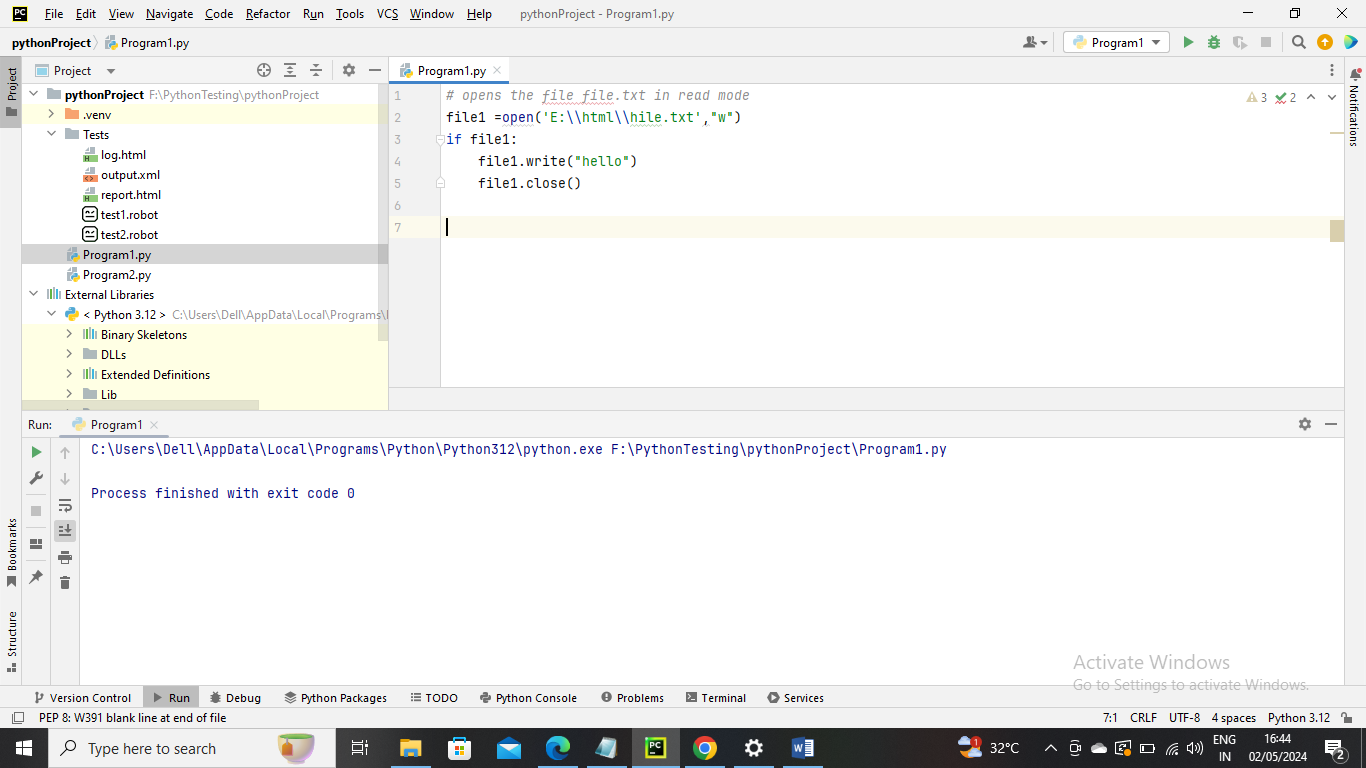


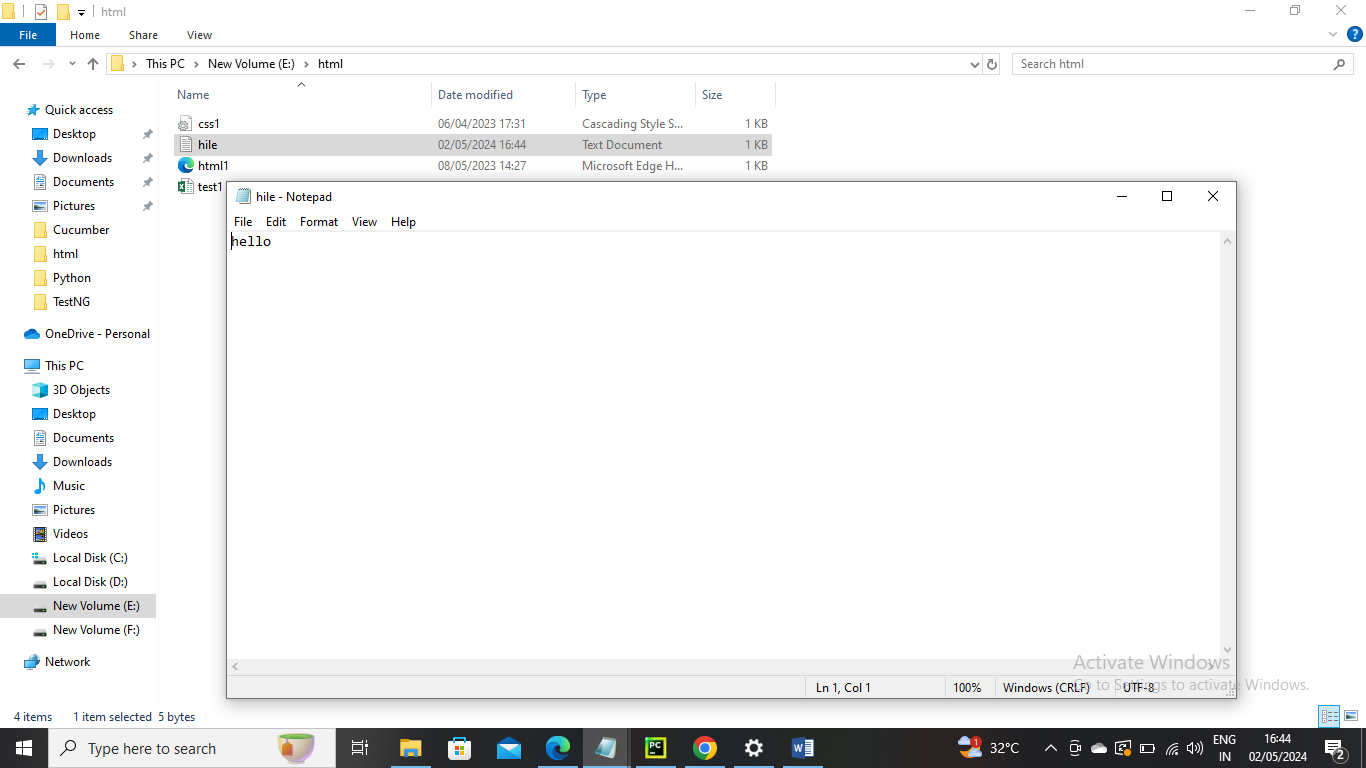
close() Method

the close() method is used to close the opened file

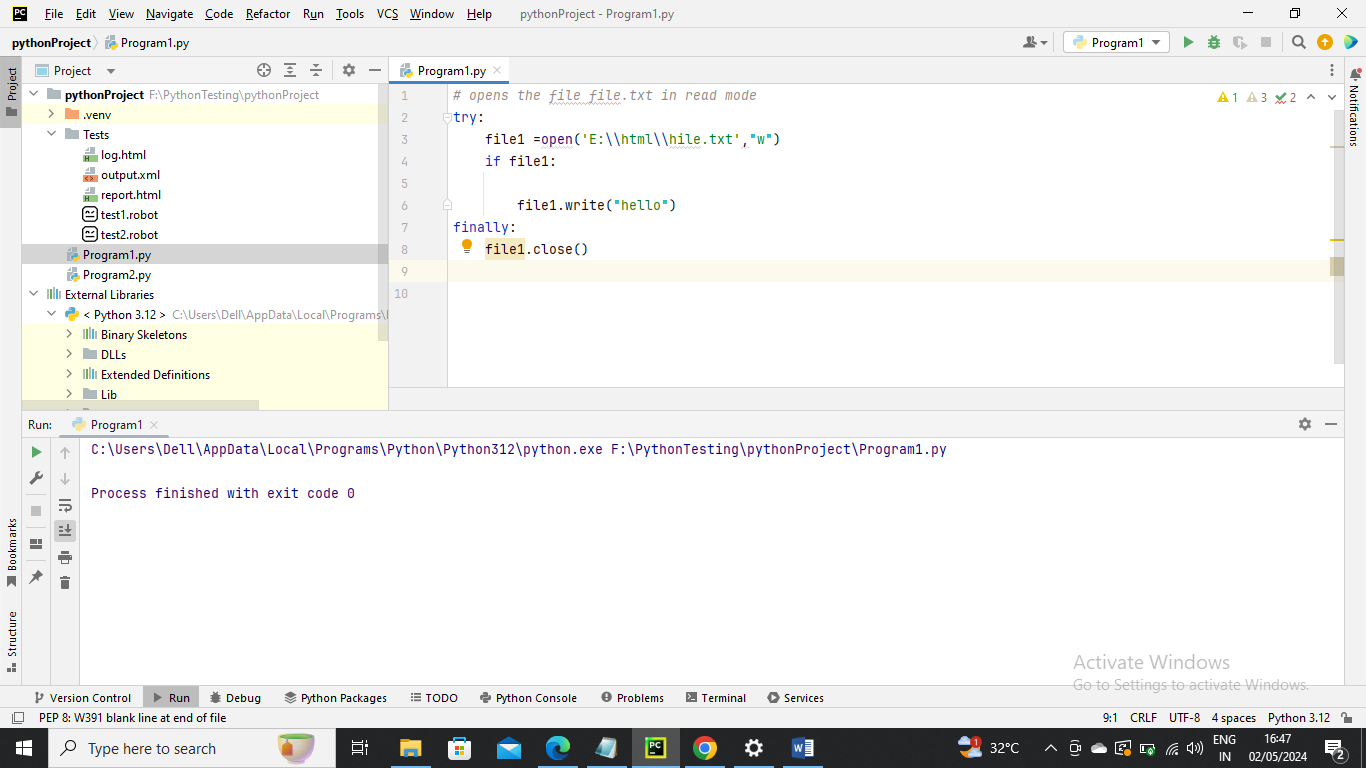
syntax – fileobject.close

Program for write mode





Another example for close



The with statement –

The with statement is useful in case of manipulating the files. It is used in the scenario where a pair of statements is to be executed with a block of code in between

Syntax-

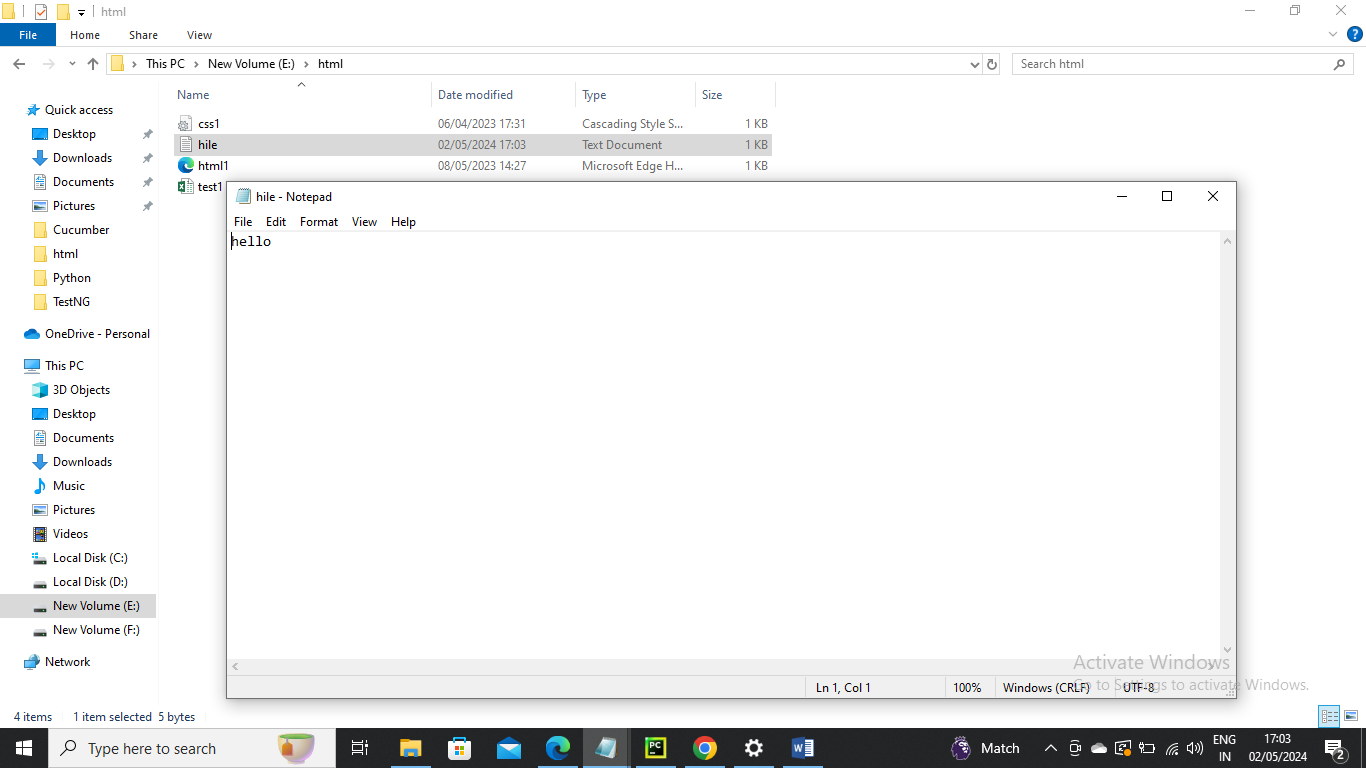
With open(<file-name>,<access-mode>) as <file-pointer>):

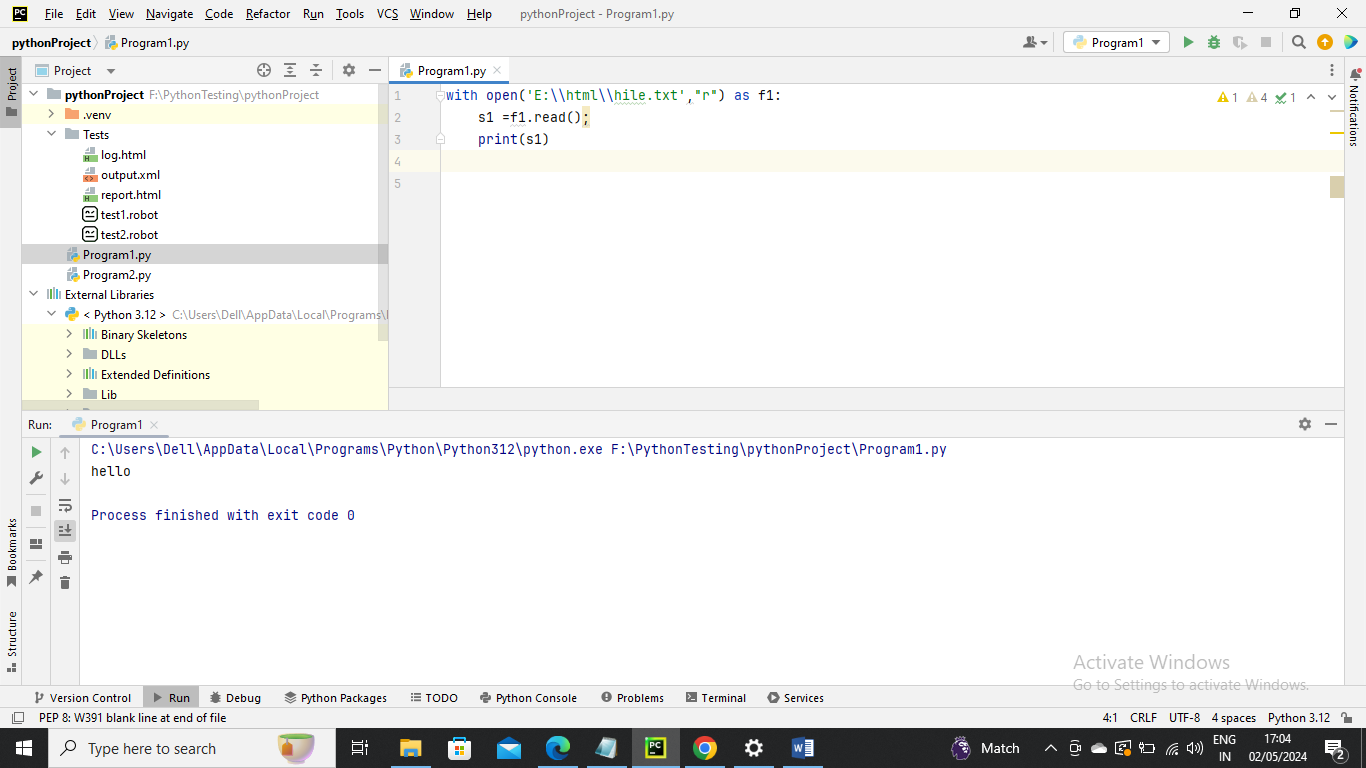
Statements

The advantage of using with statement is that it provides the gurantee to close the file regardless of how the nested block exists

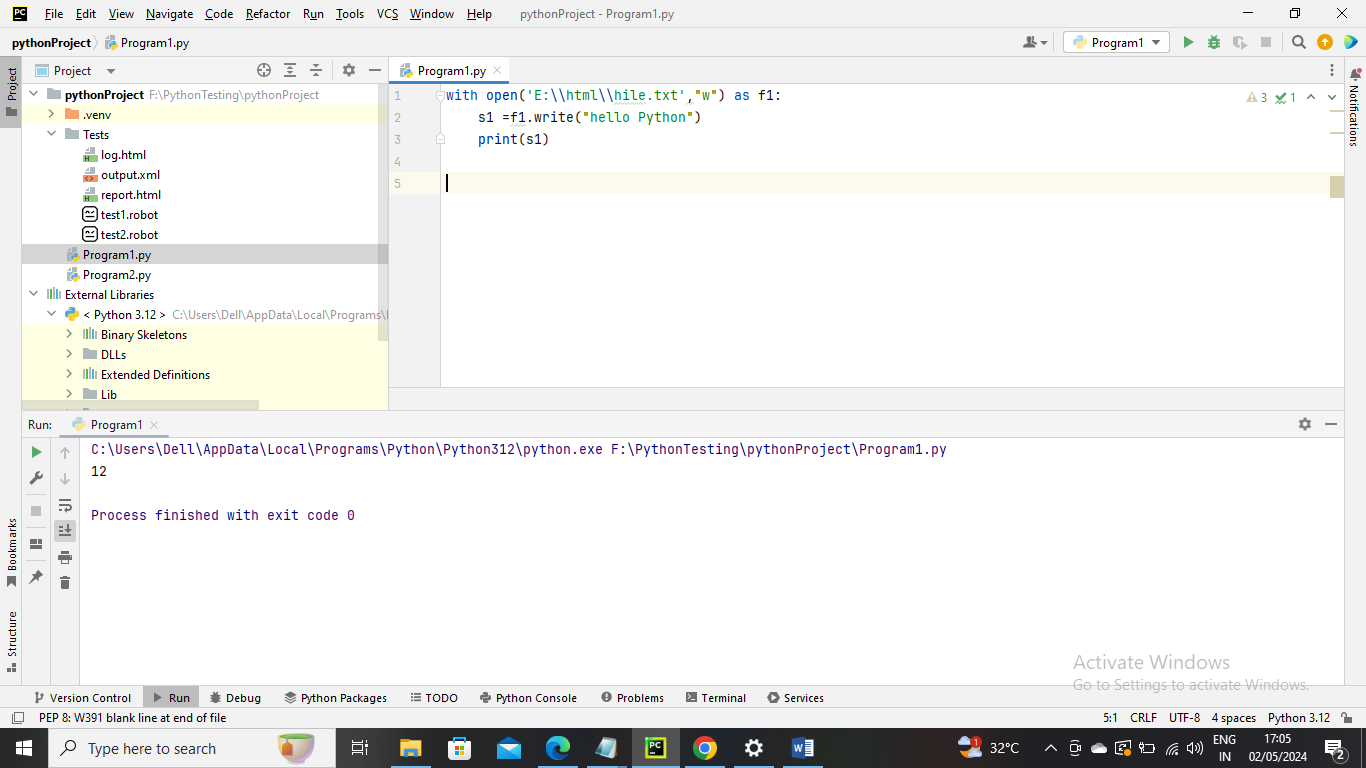
It is always suggestable to use **with** statement in the case of files because, f the break,return or execption occurs in the nested block of code then it automatically closes the file, we don’t need to write the close() function. It doesn’t let the file to corrupt

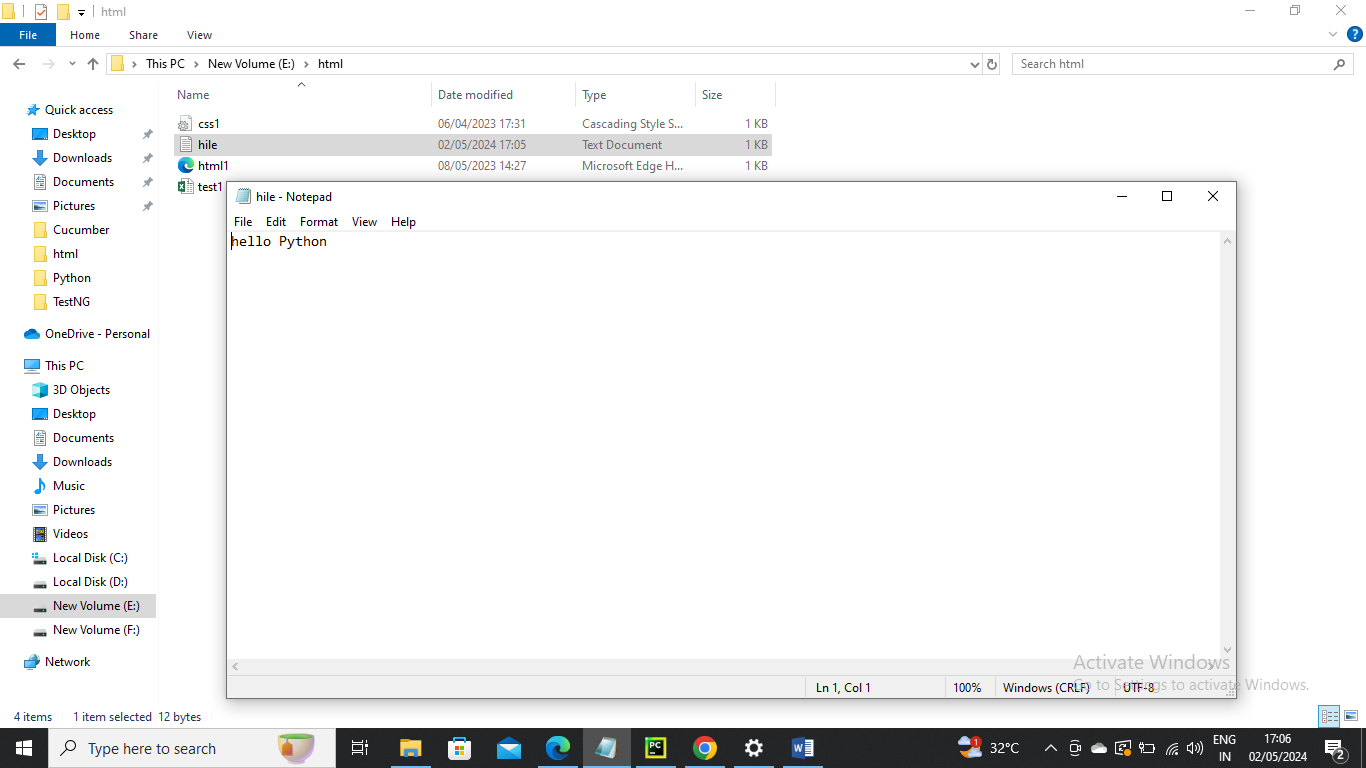
Example1





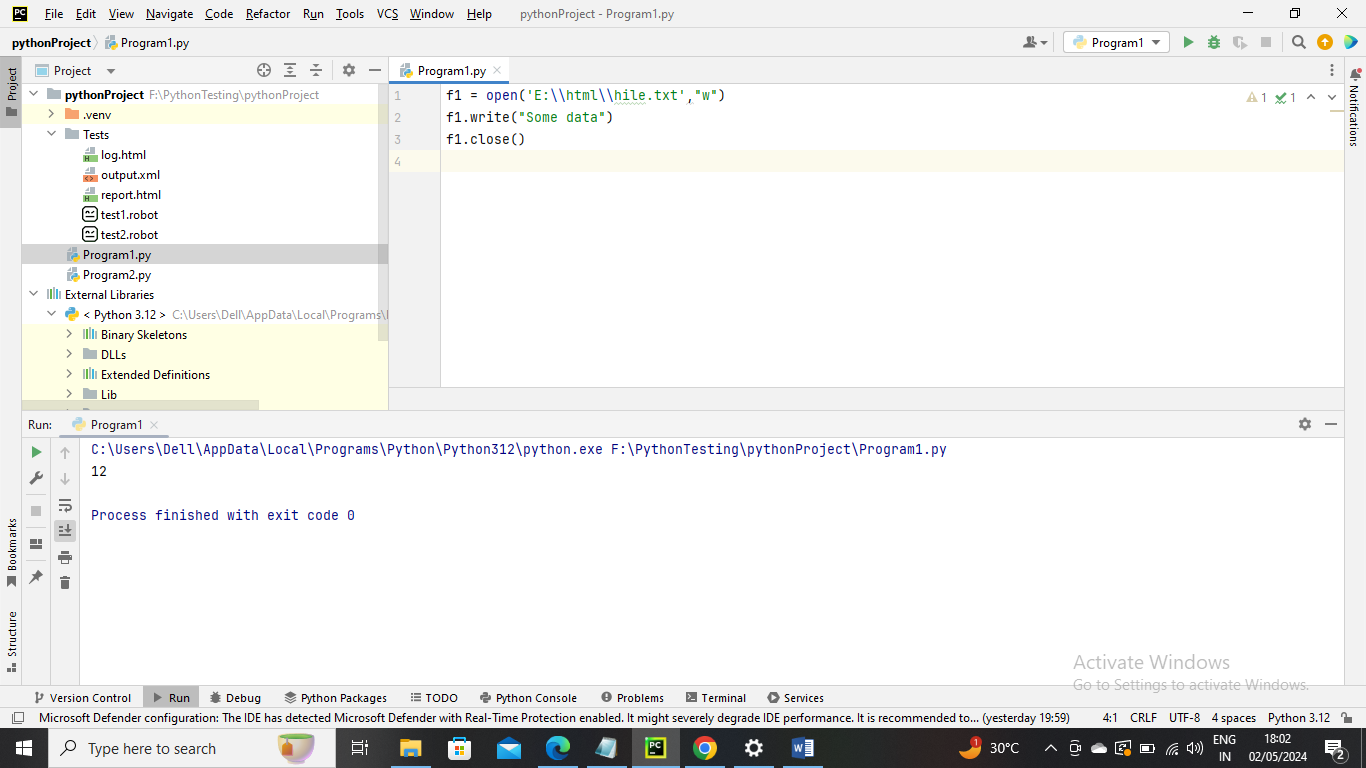
Example 2



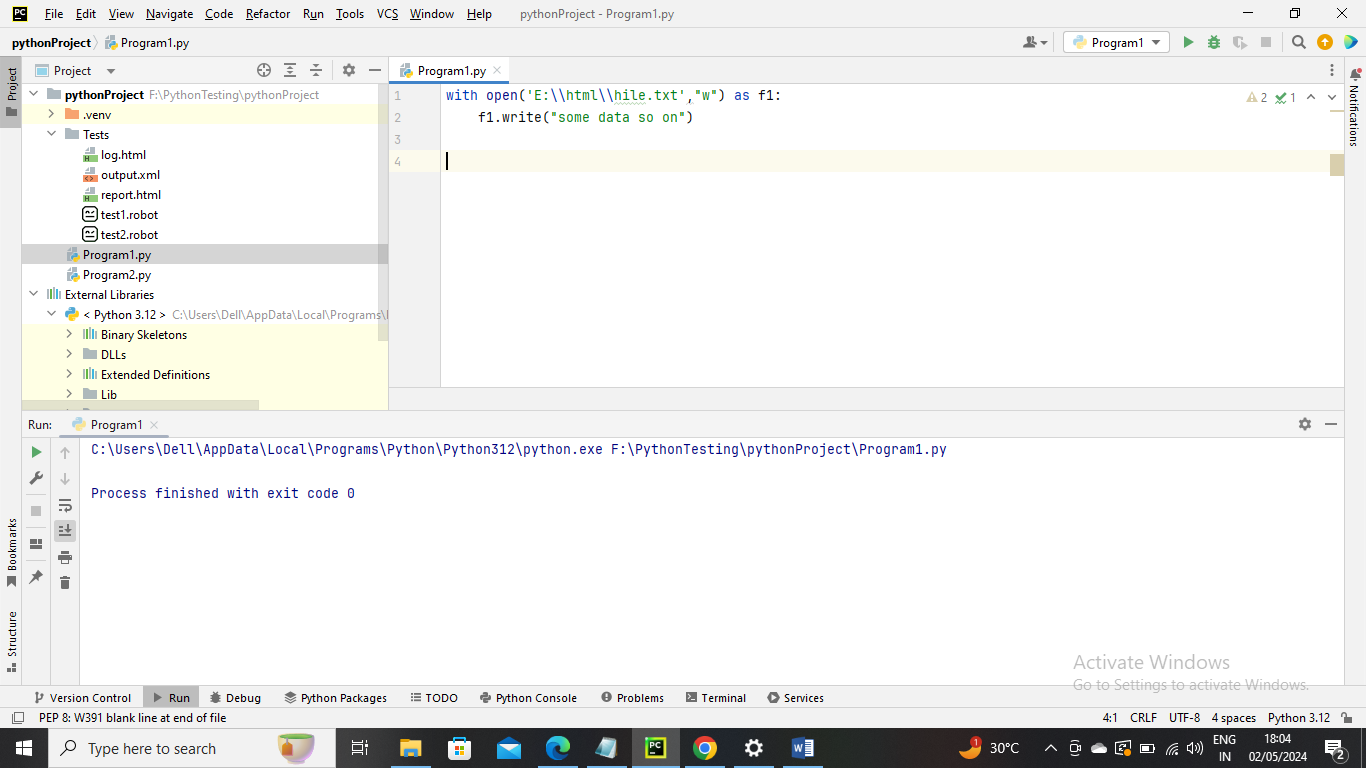


Writing the file –

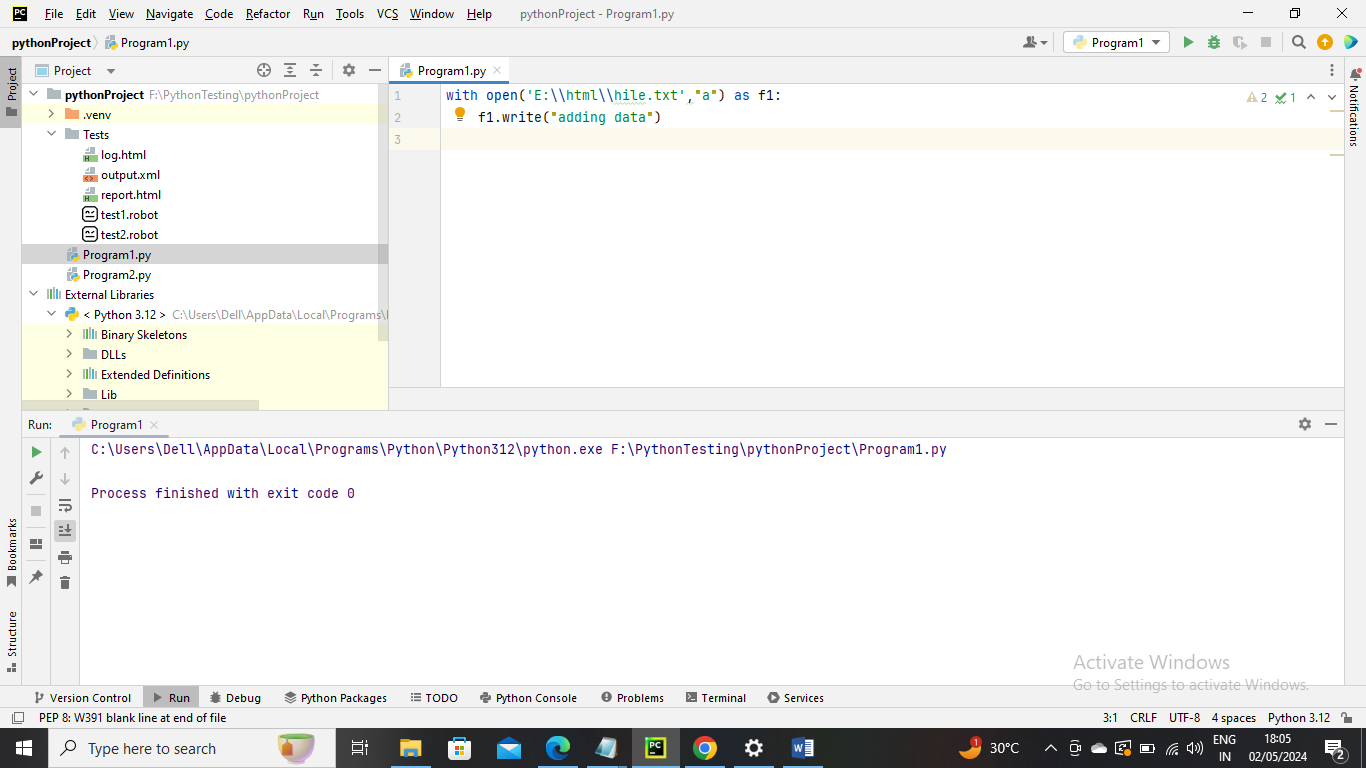
Example 1

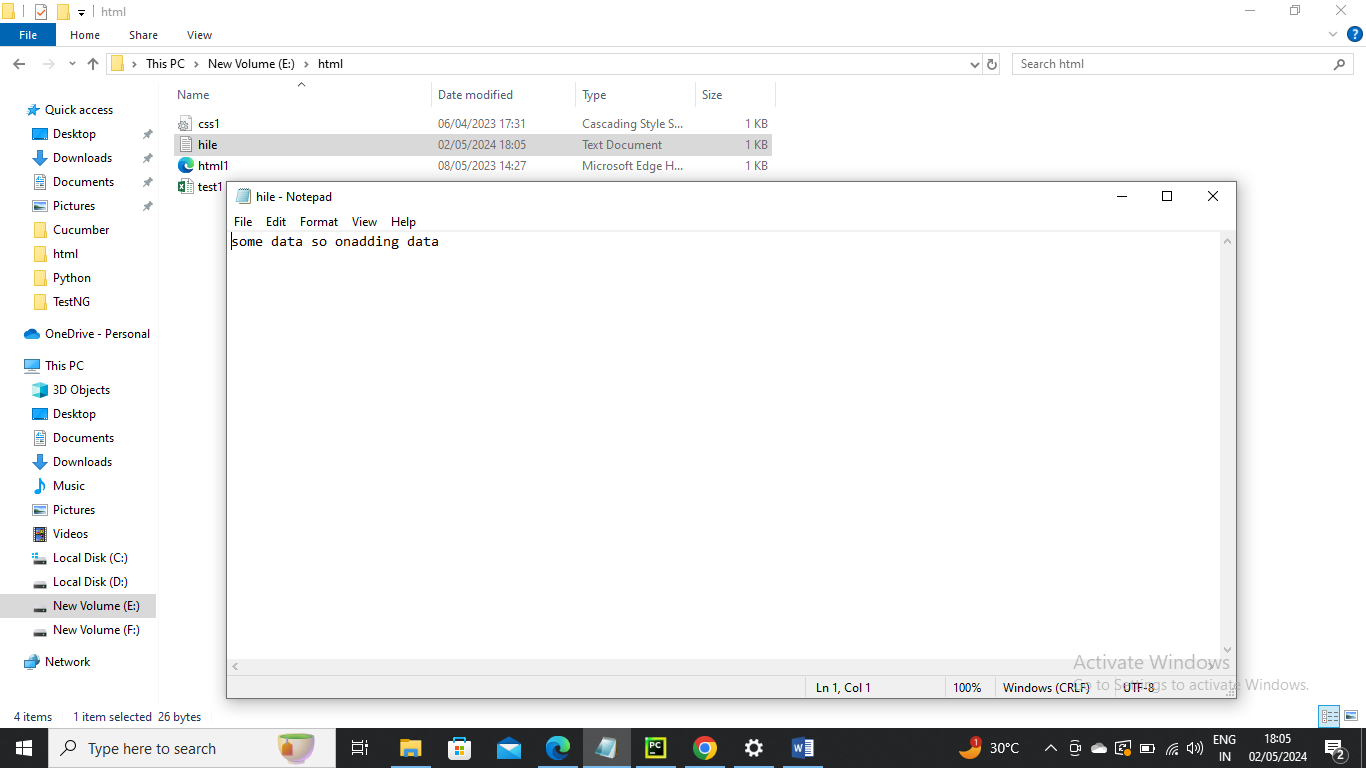


Example 2



Example 3 append

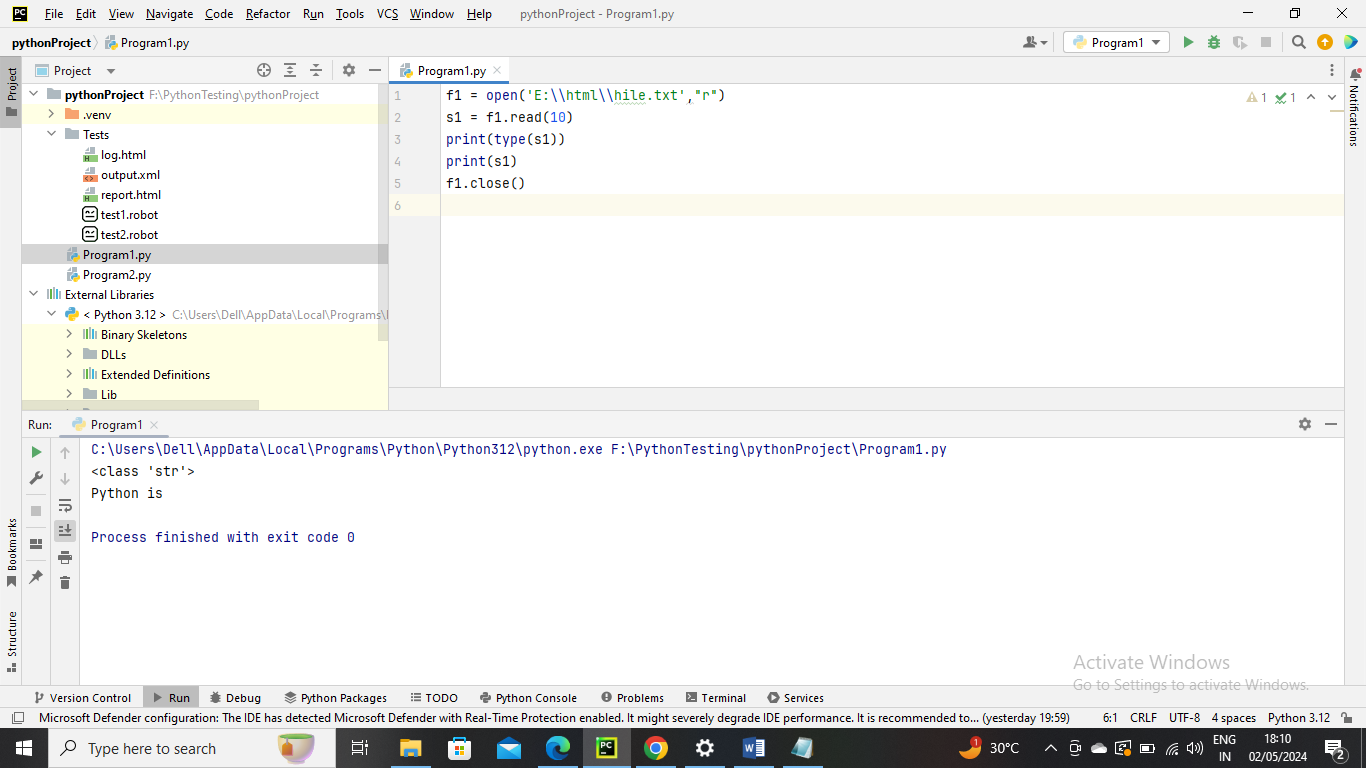




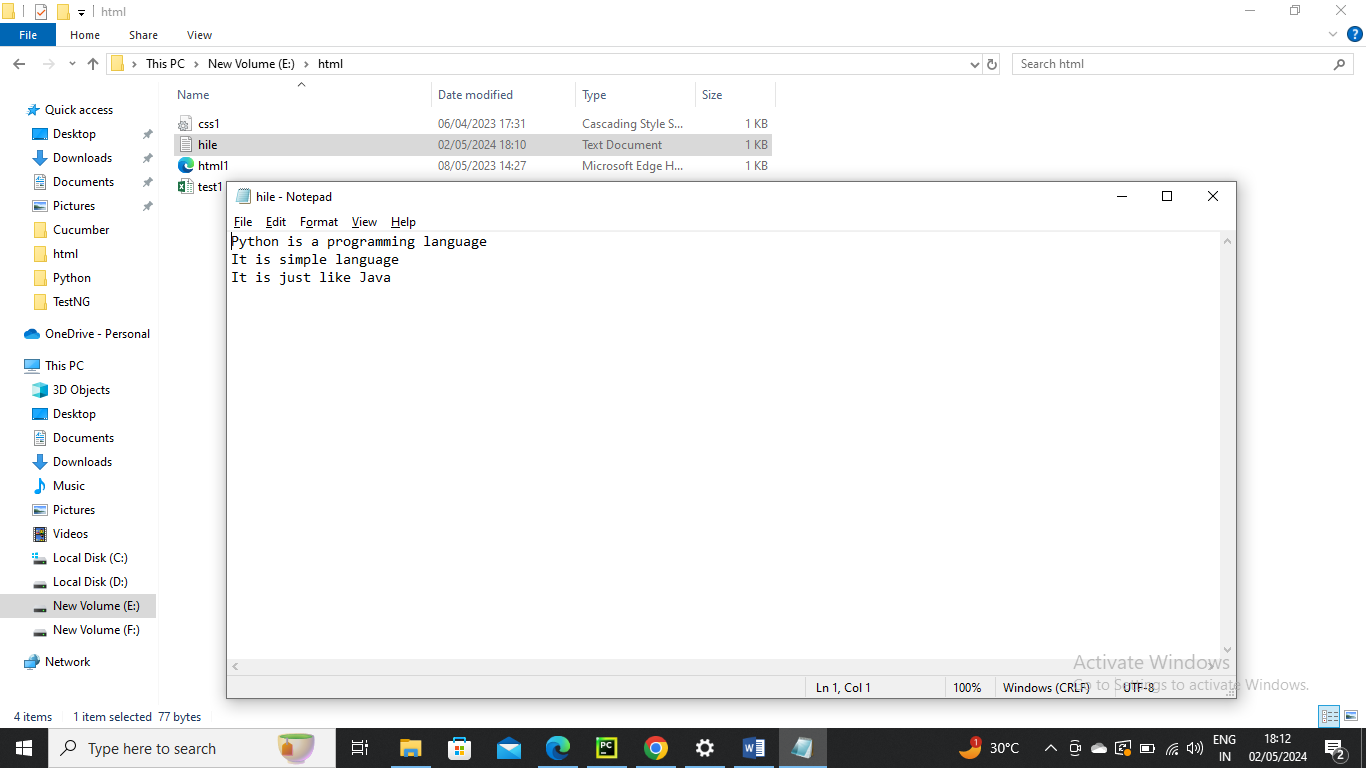
Reading a File

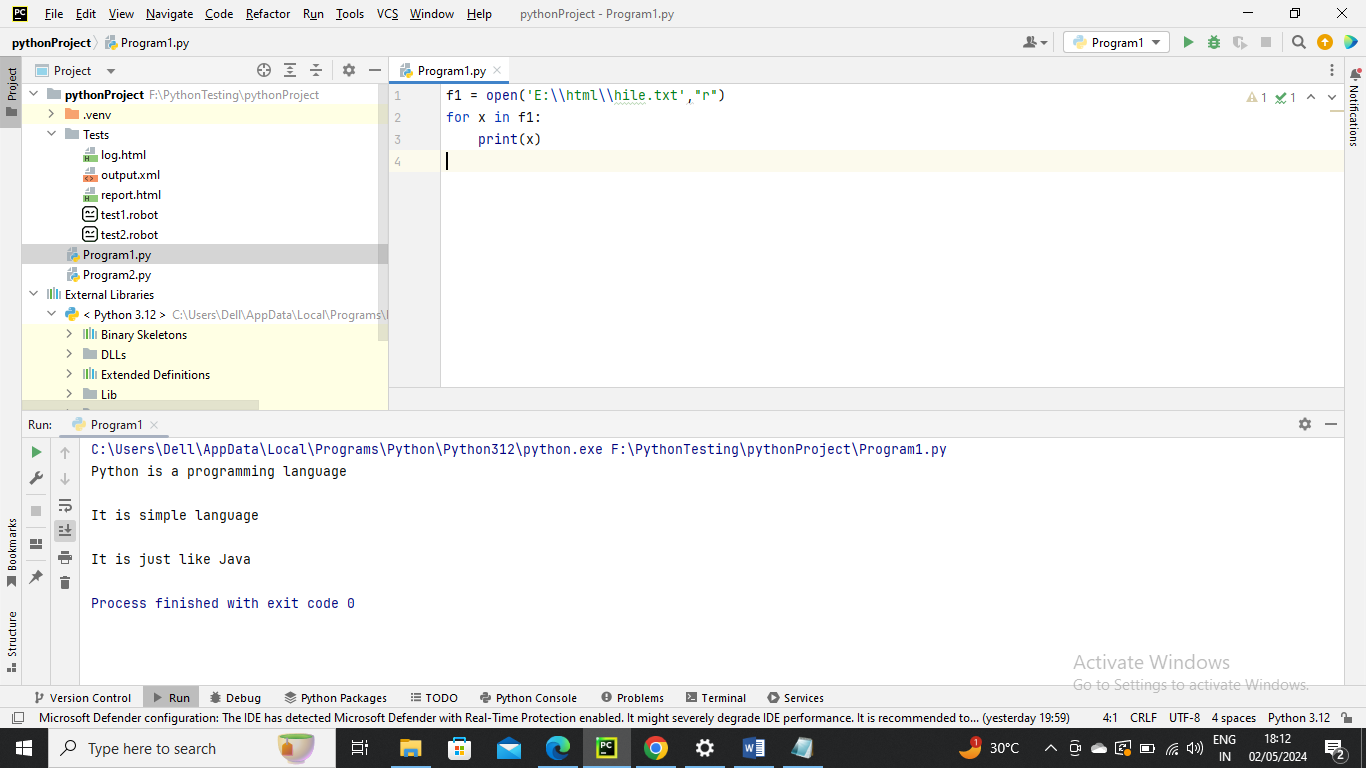
Data in file



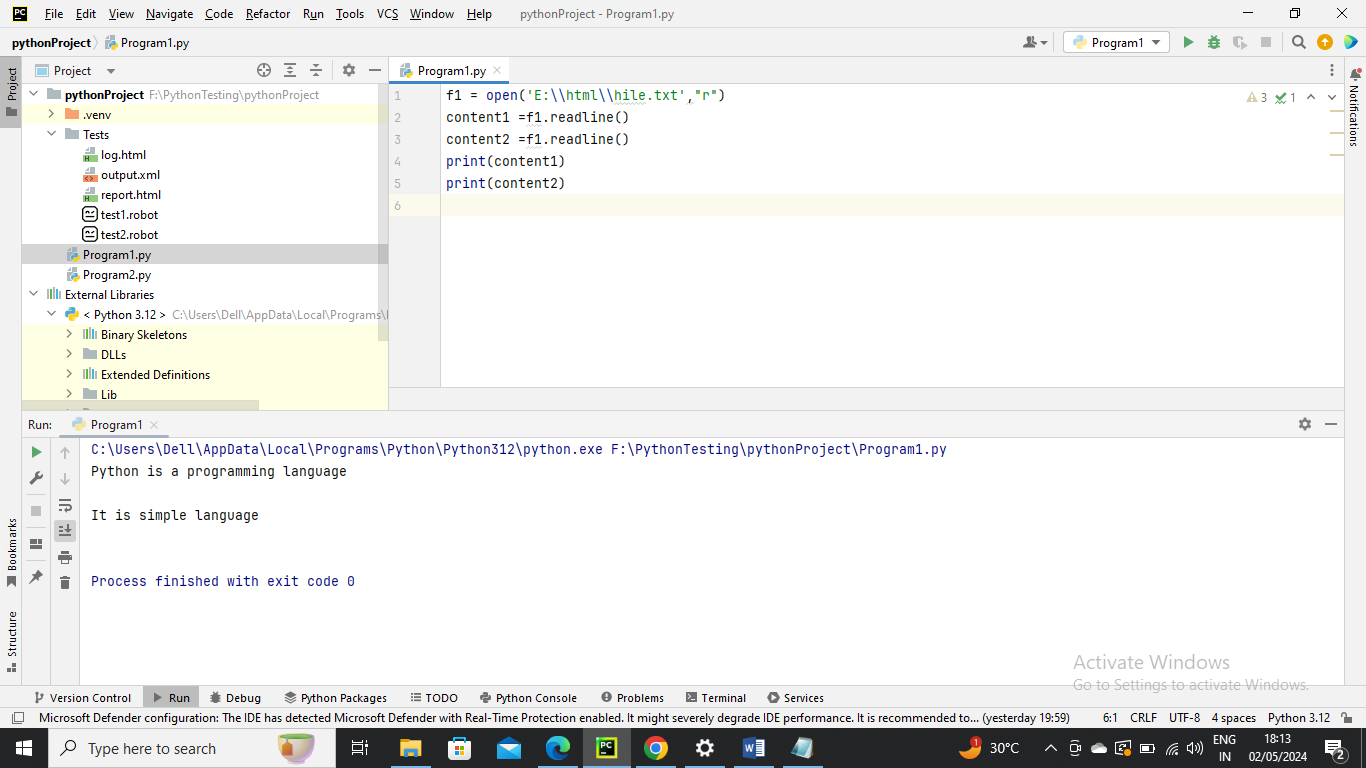


Example 2 – read file using For Loop to read each line until the end

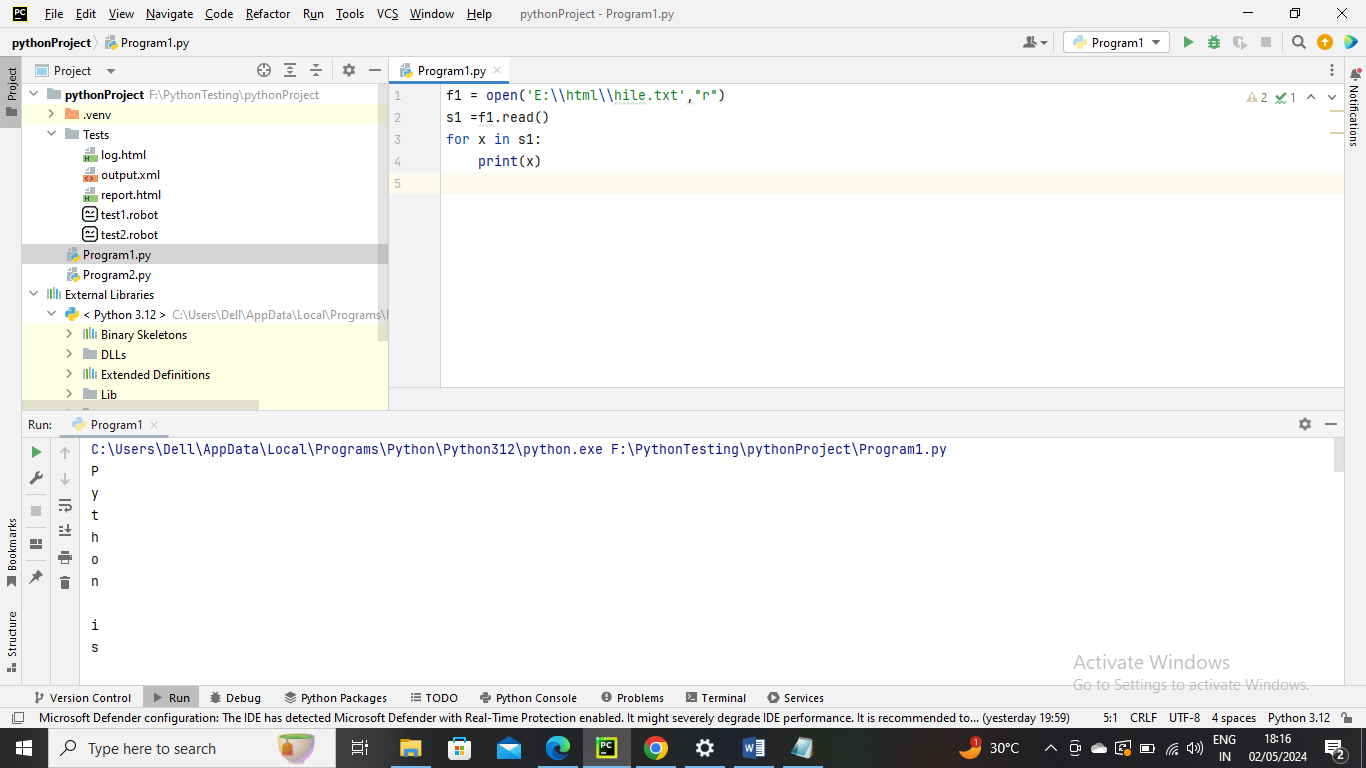


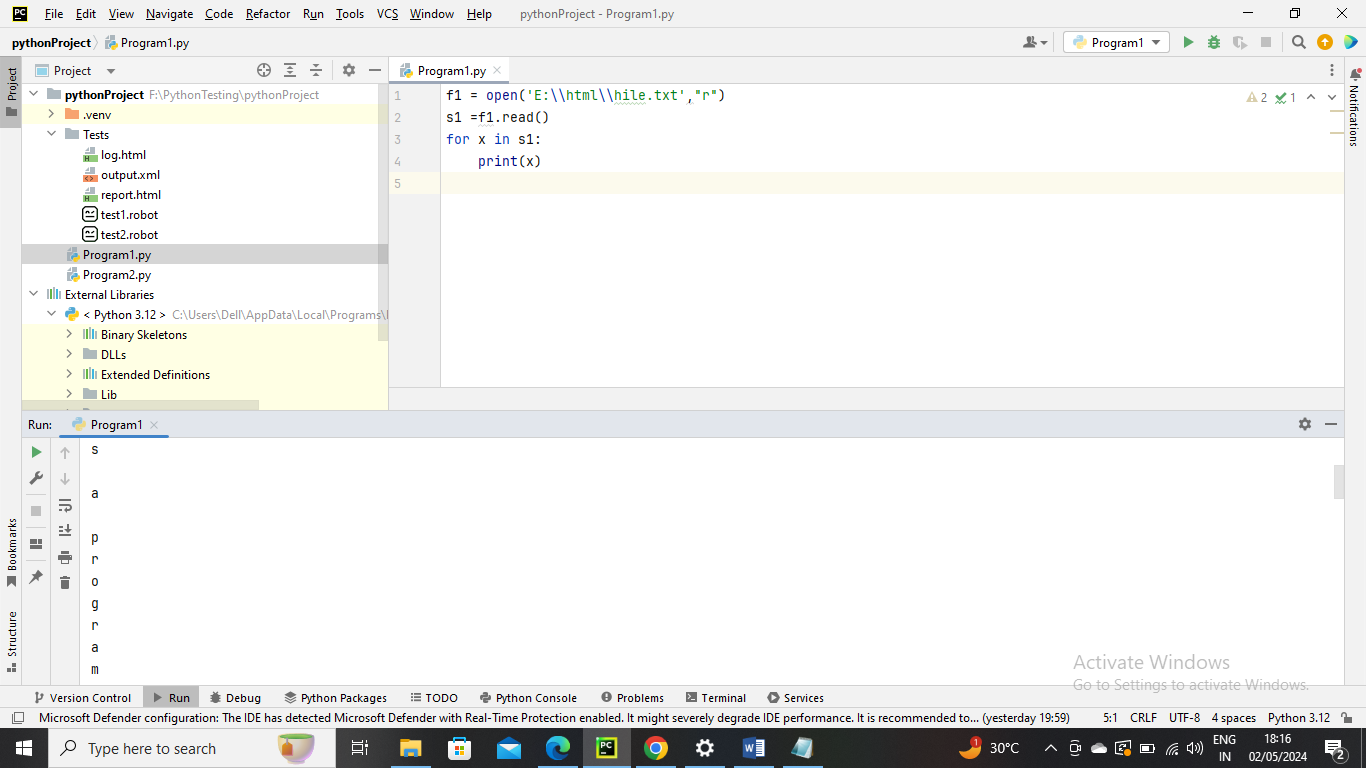


Another way to read each line

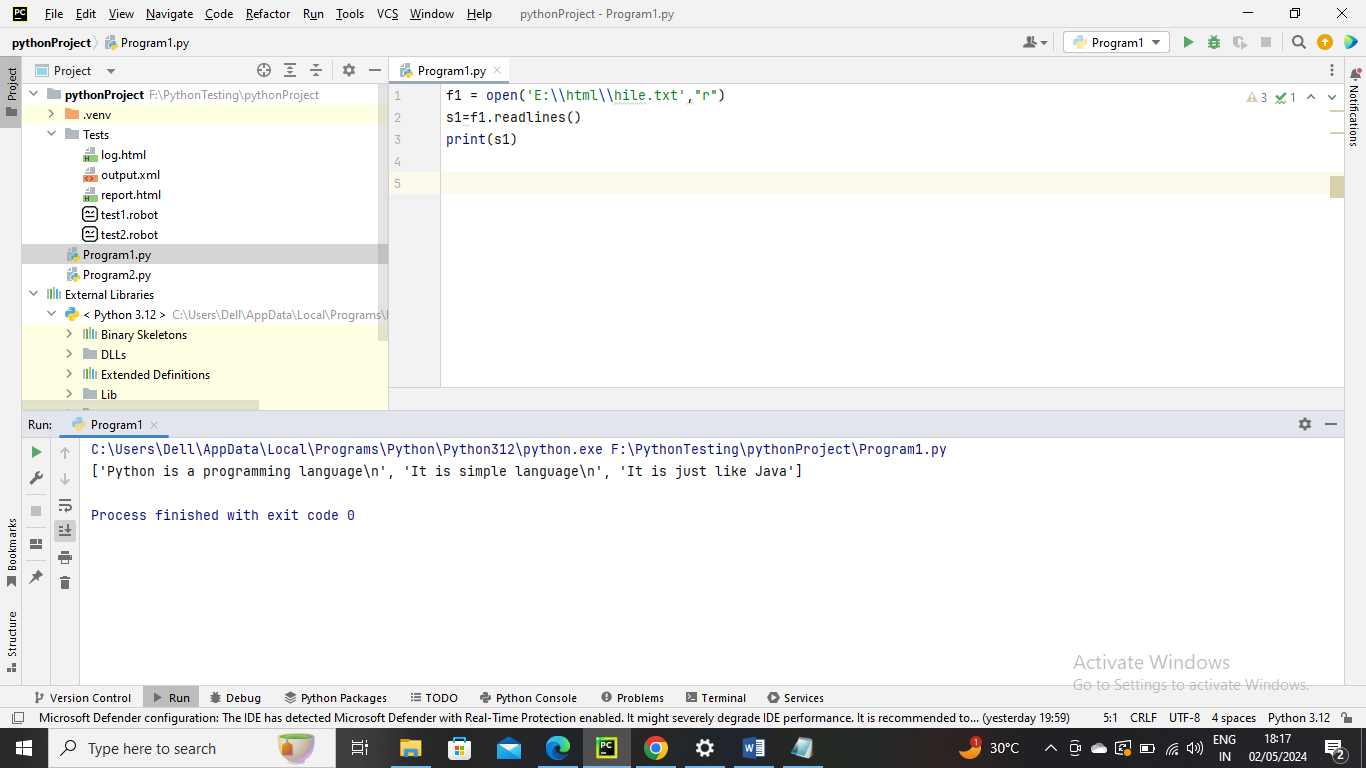


Read file using for loop to get each letter in file





Example – Another way to read all the line in a file



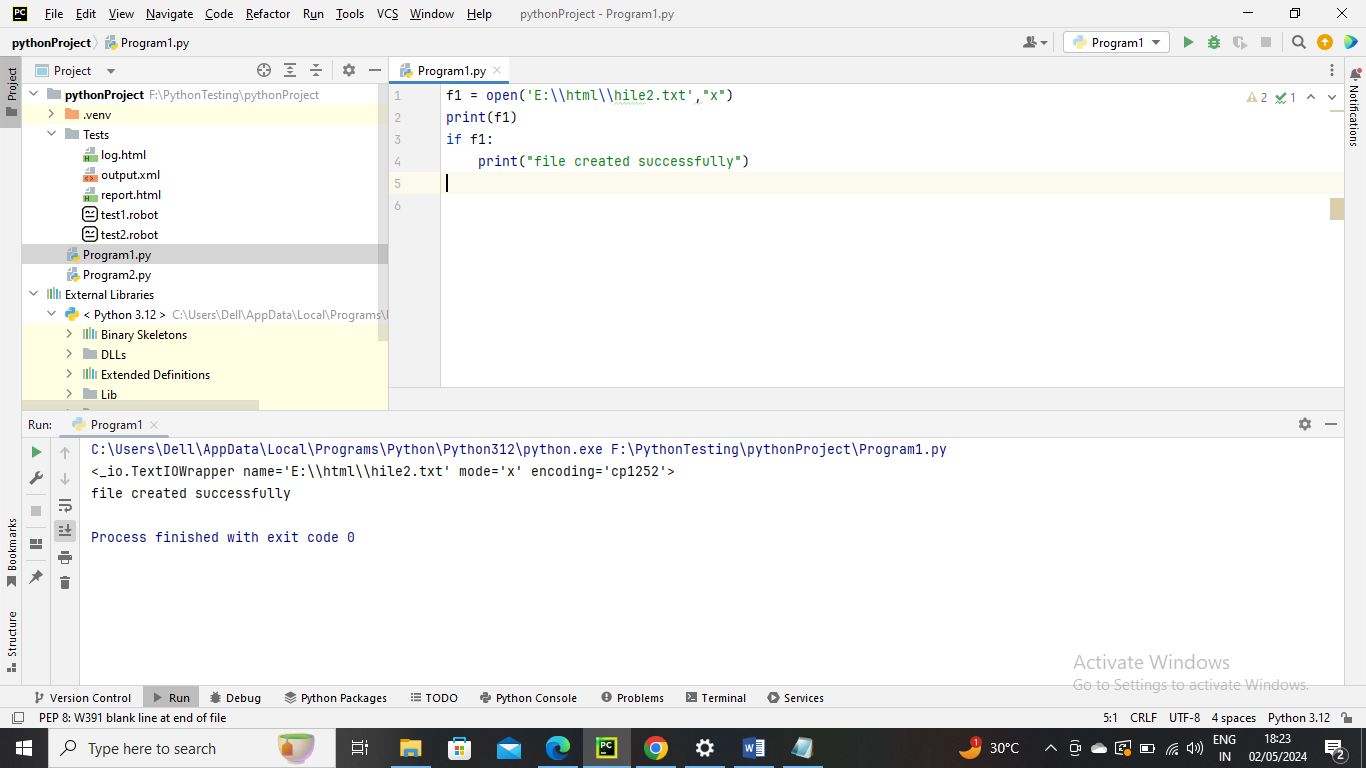
Creating a new file –

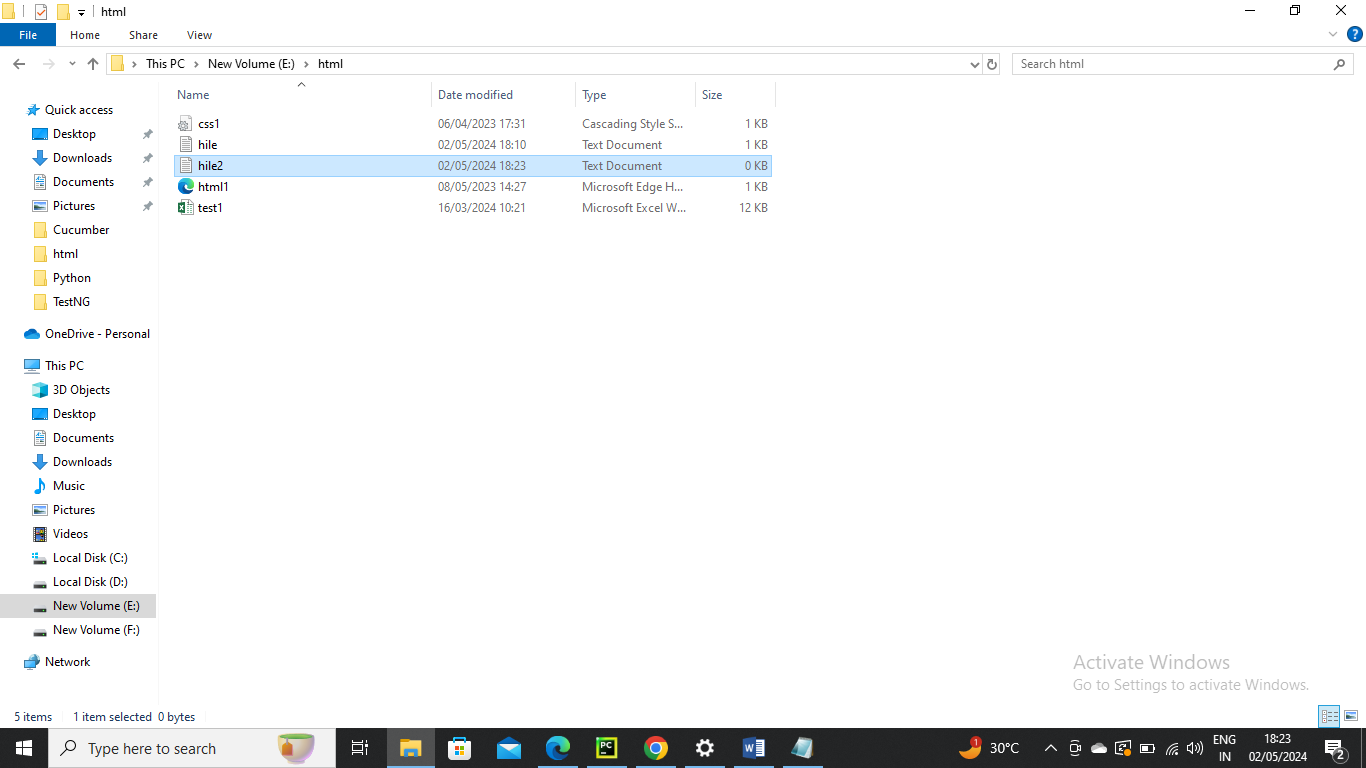
x – it creates a new file. It causes error if file exists with same name

a – it creates a new file with specified name if no such file exists. It appends the content to file of the file already exists with specified name

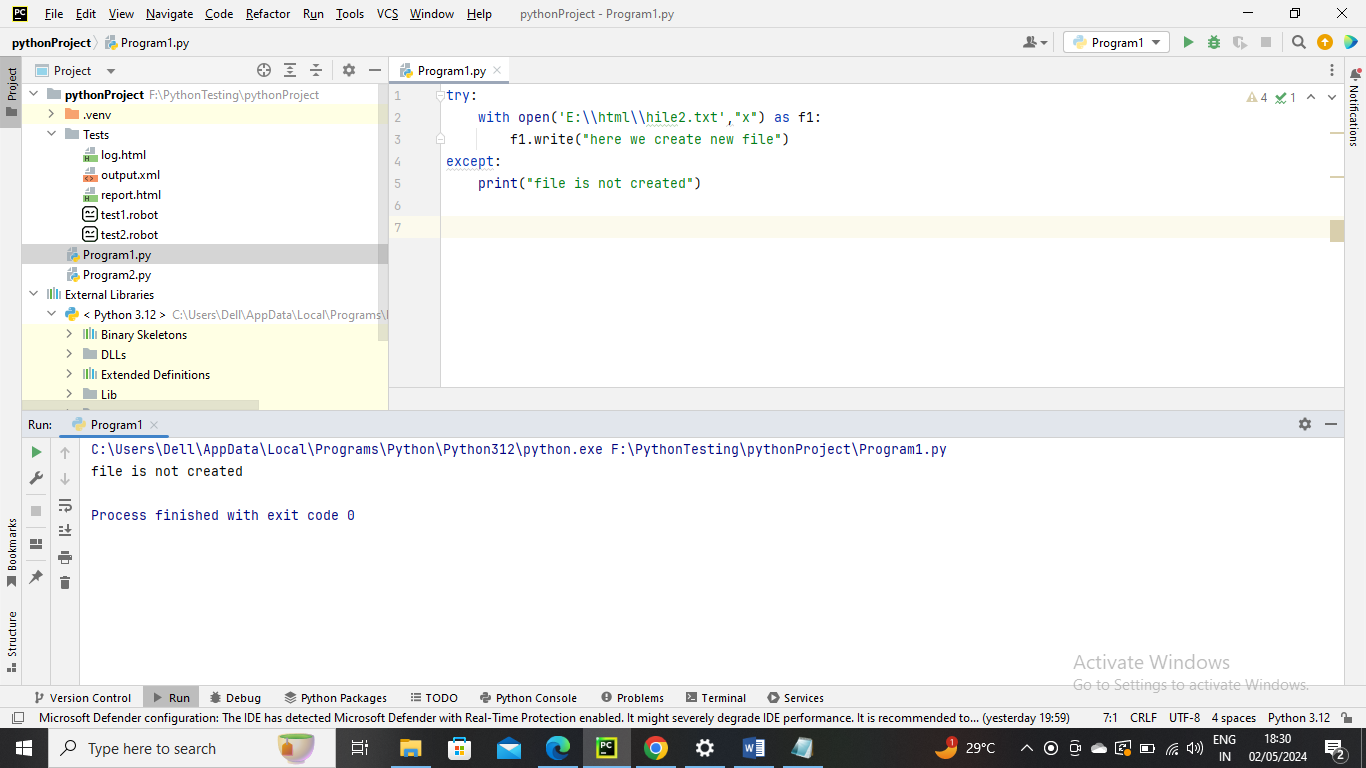
w ---- creates a new file with specified name if no such file exists. It overwrites the existing file

Example1 – creating new file

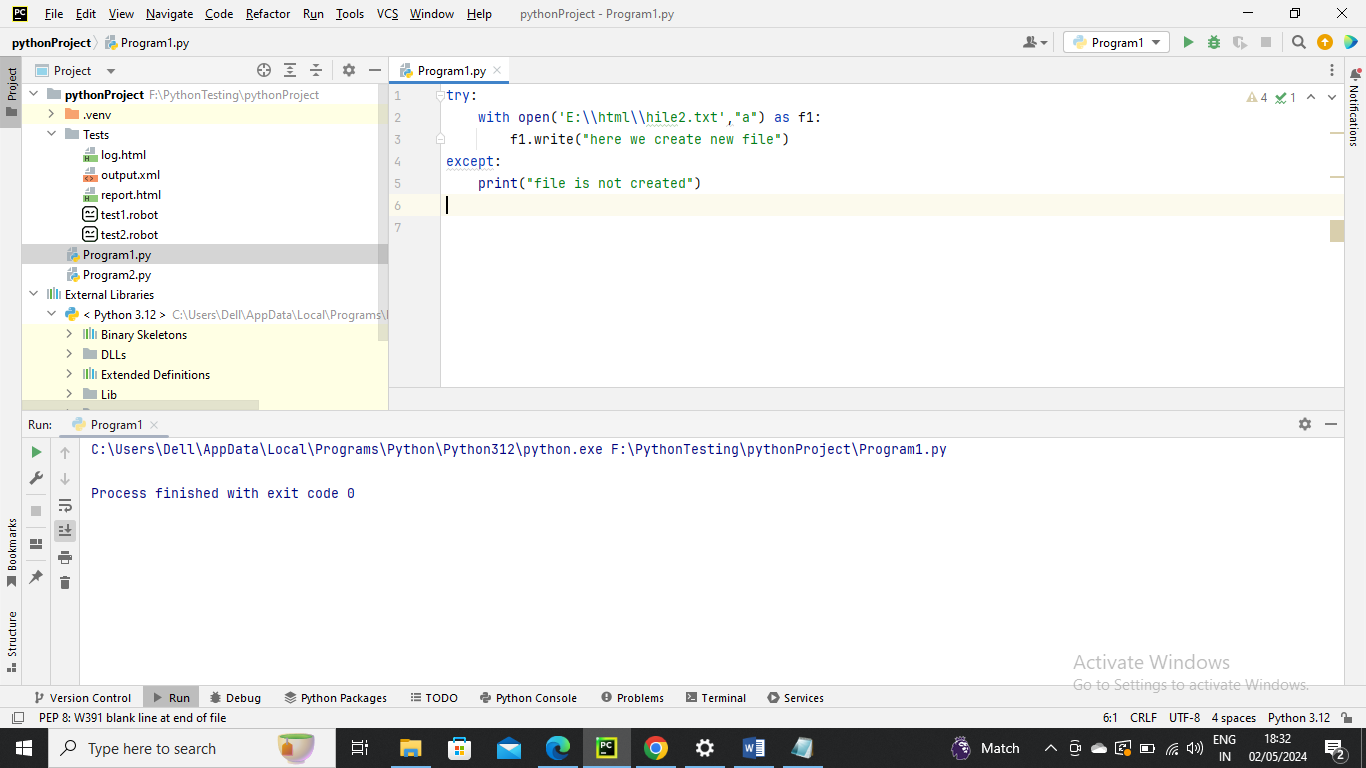


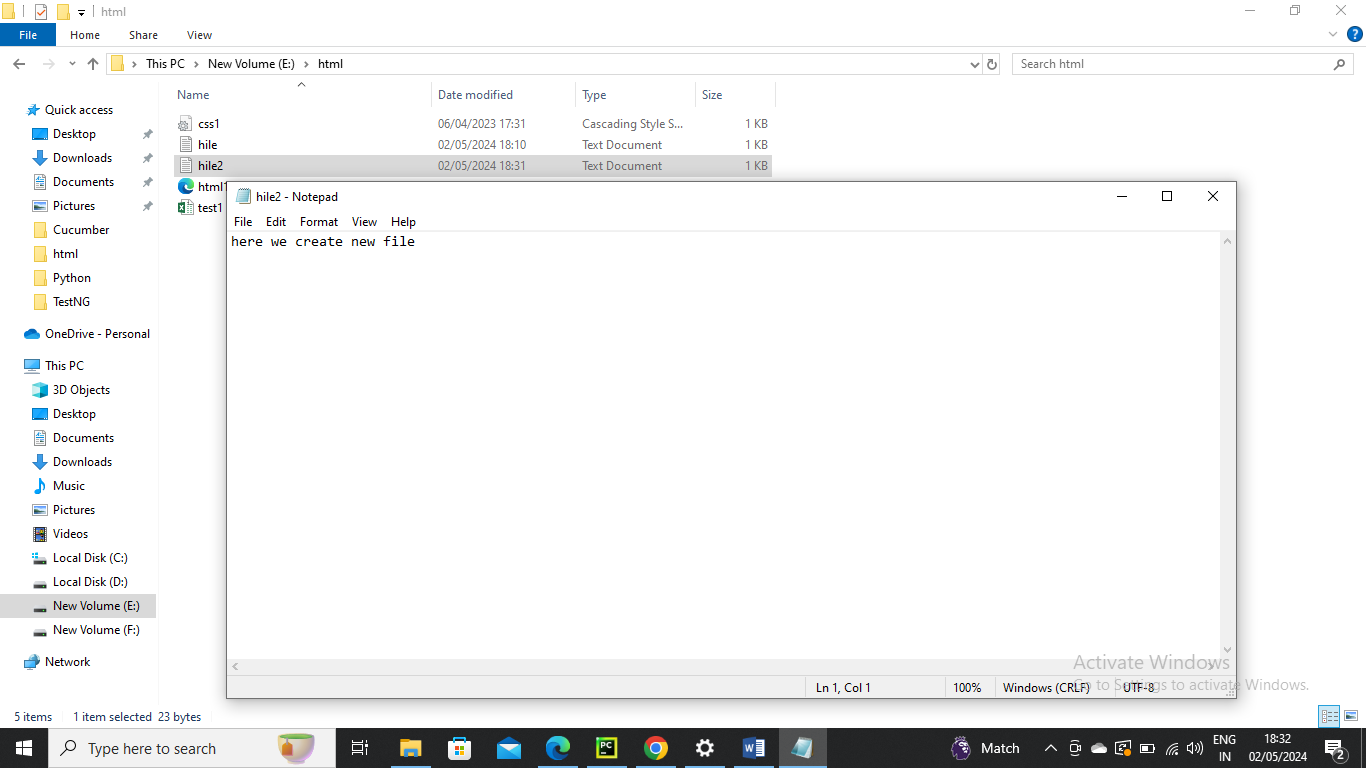


Example 2 – Since file is already created it is thrown error saying file not created via exception

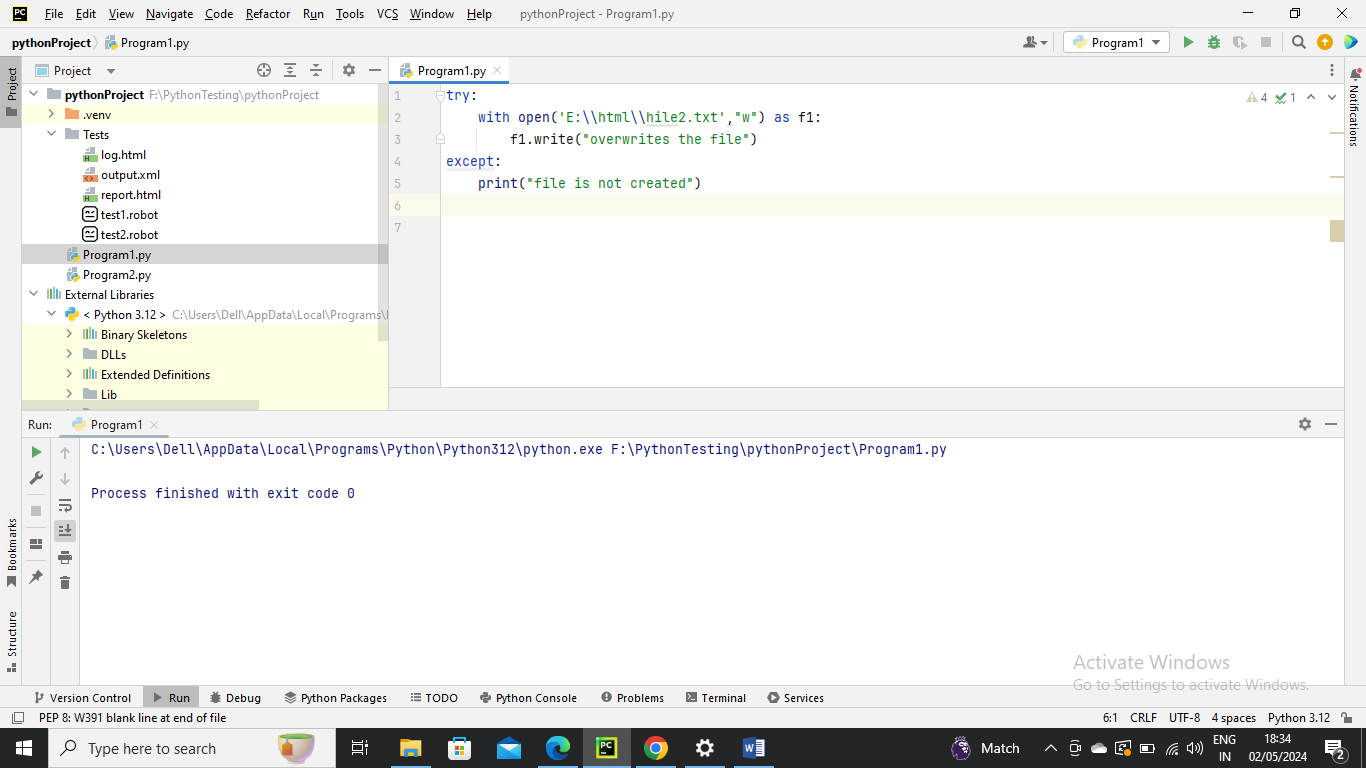


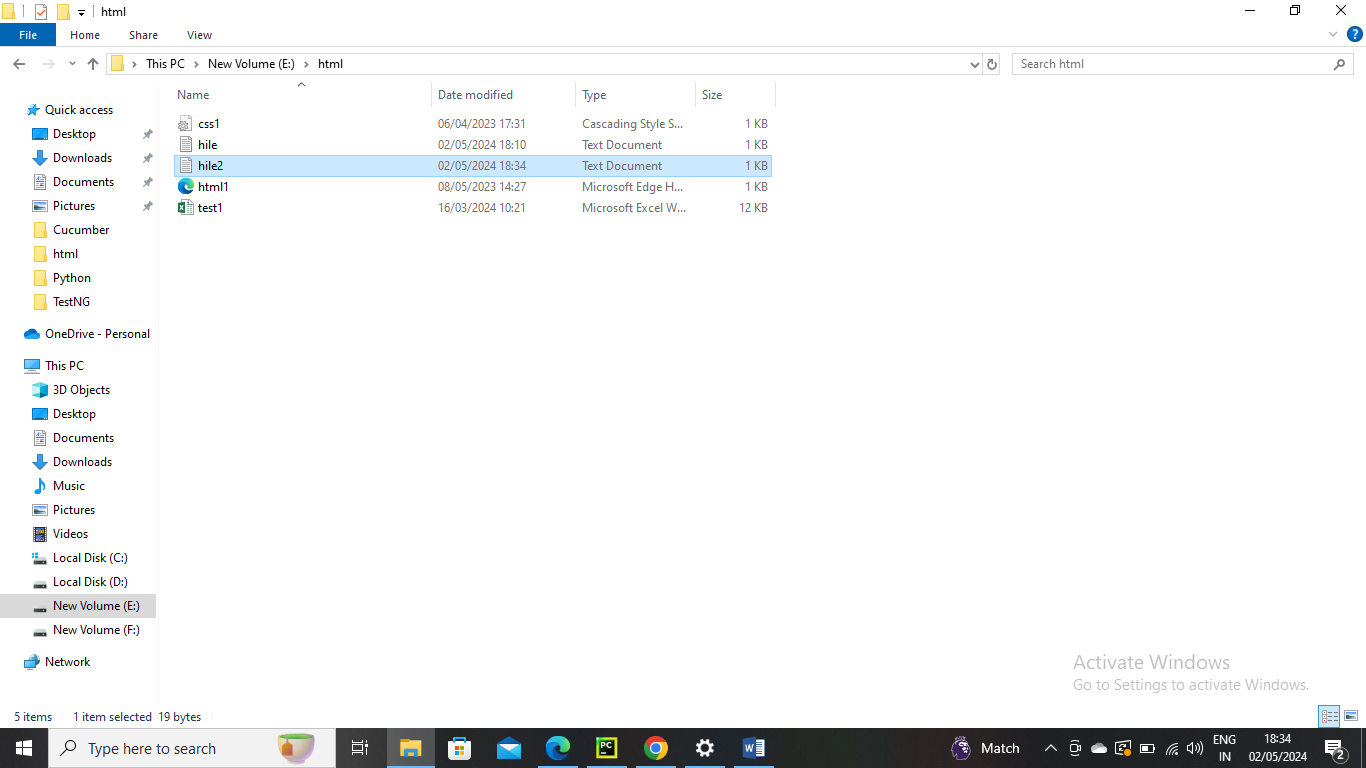
Example 3 --- here in below since hile2 file already created with access mode a it will append the data to existing file





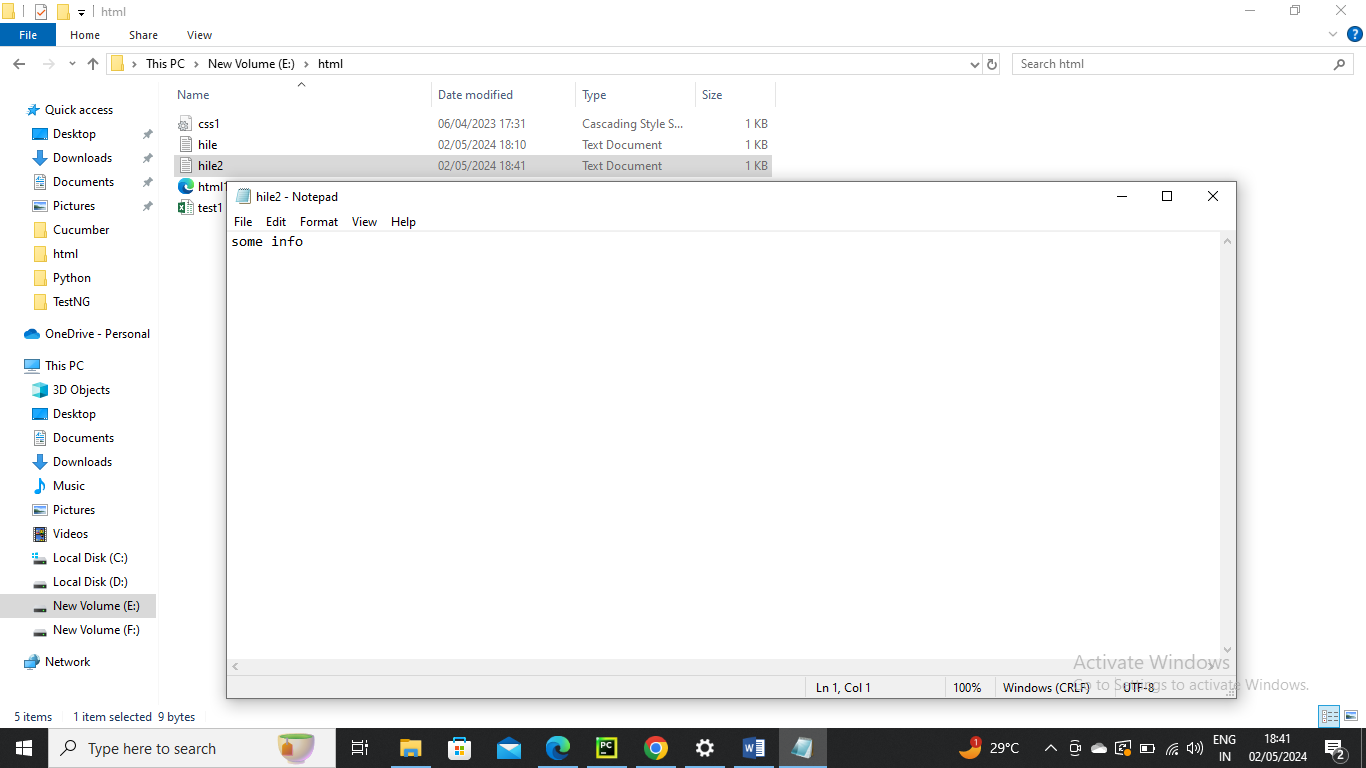
Example 3--- with access mode “w” it will be overwrite (replaces or deletes and create new one) file

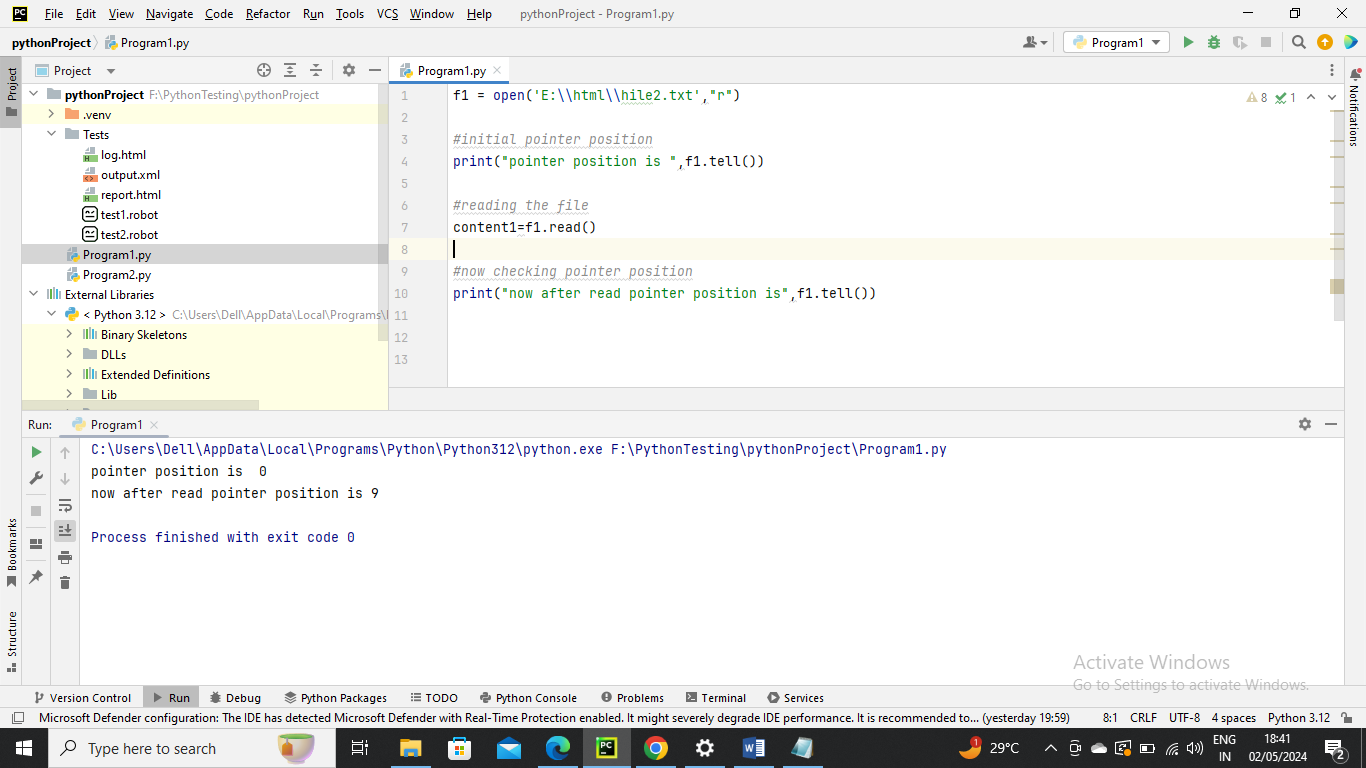




File Pointer –

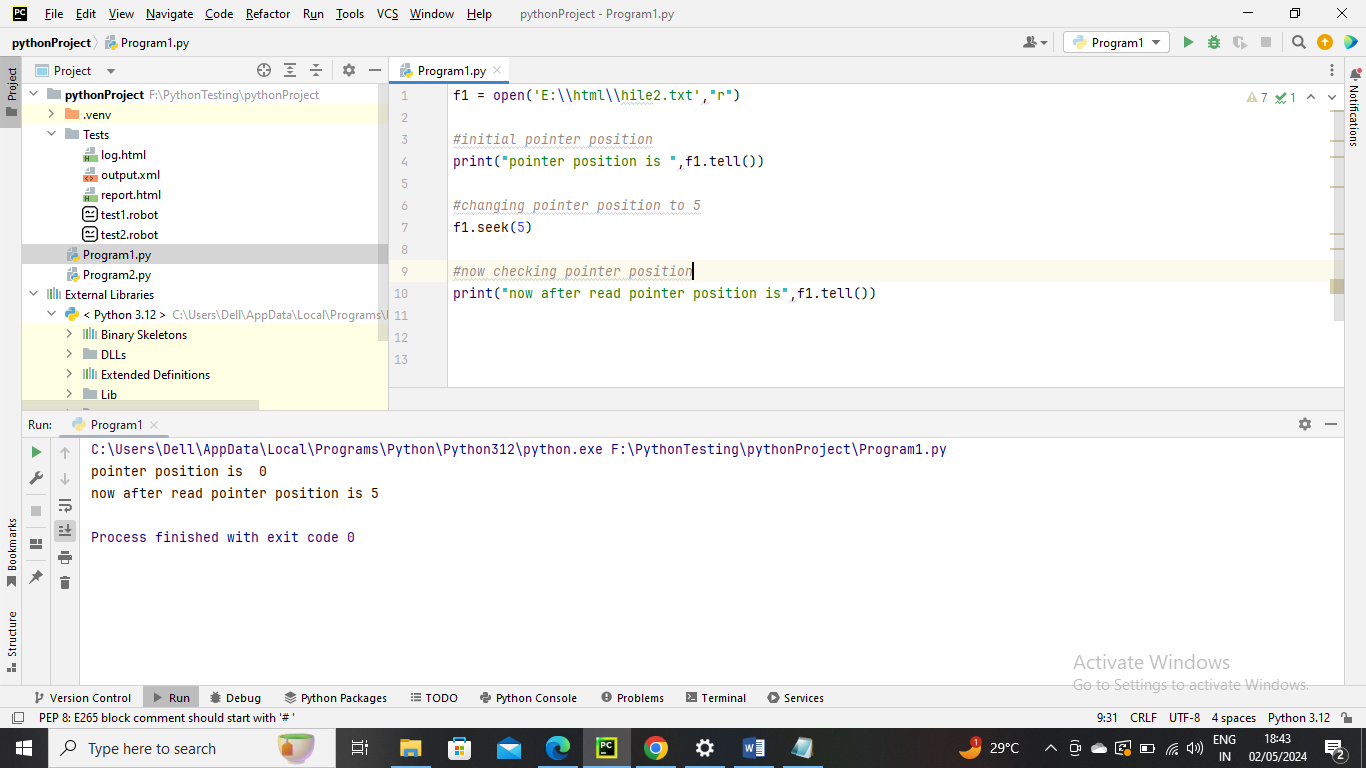
tell() method which is used to print the byte number at which the file pointer currently exists. The tell() pmethis is return the position of read or write position in this file





Modifying thepointer position

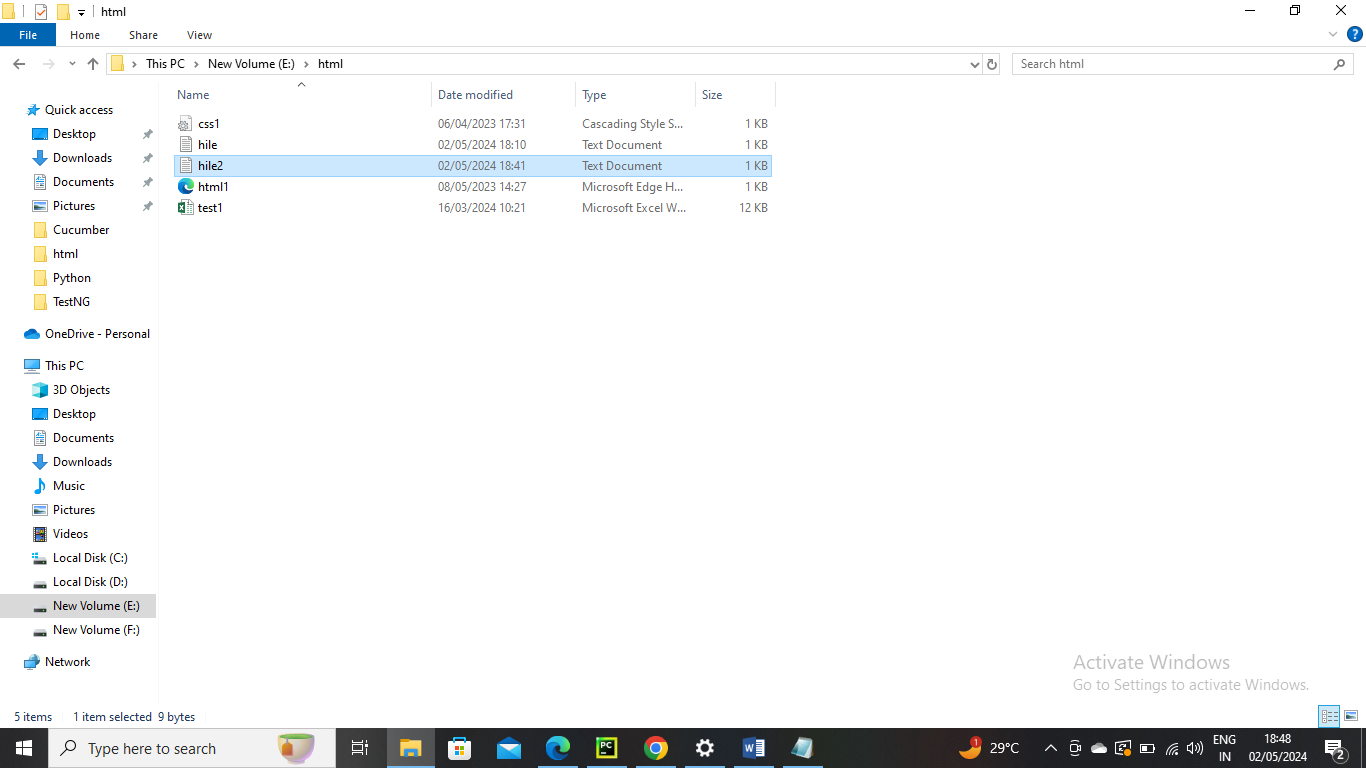
Seek() method is used to modify the pointer



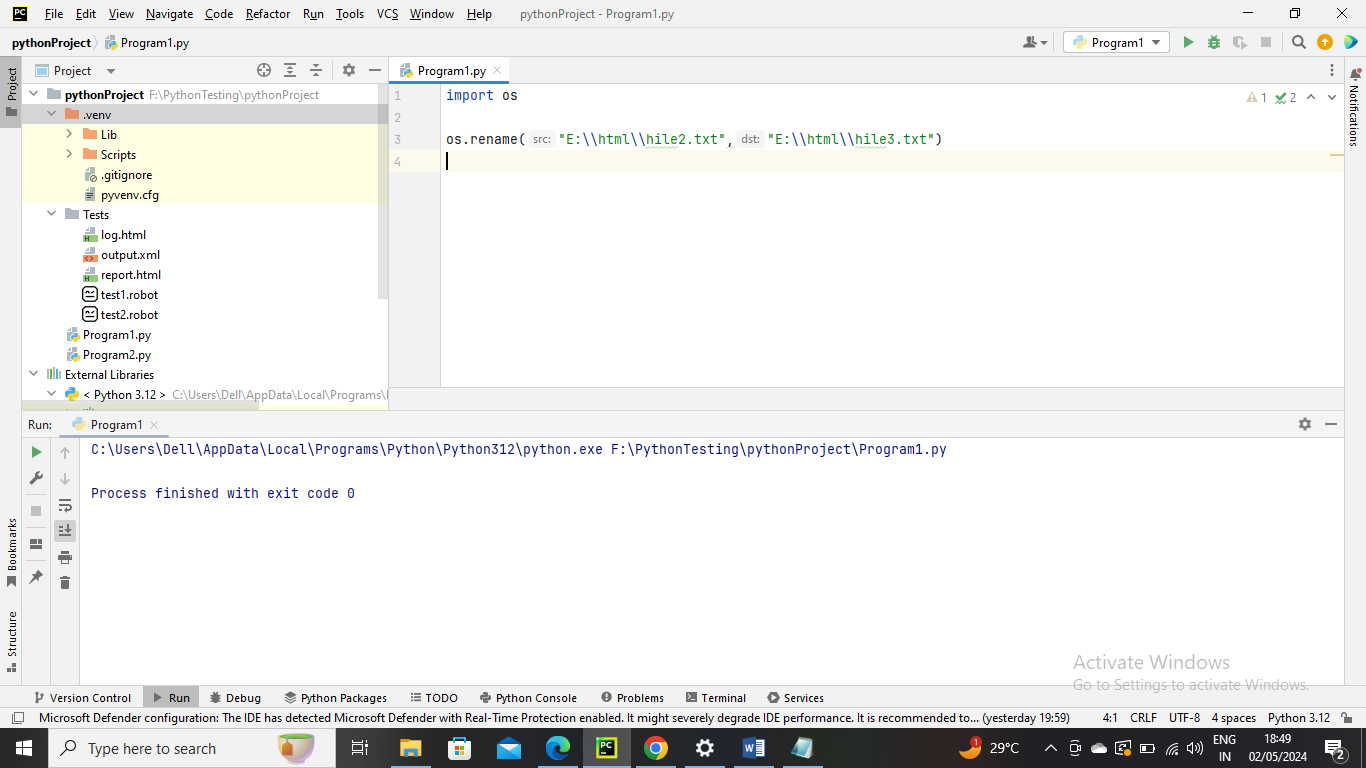
Python OS-

Renaming the file

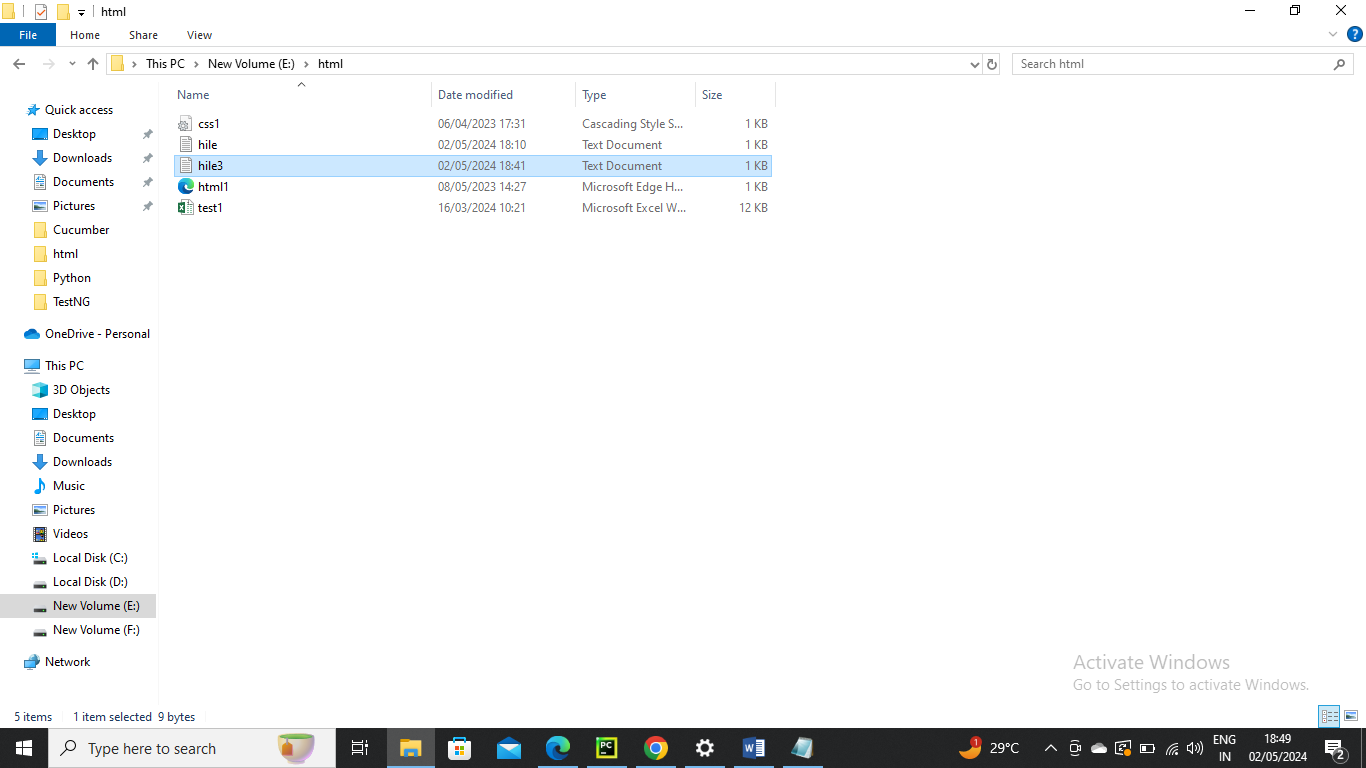
Before-



Run the program

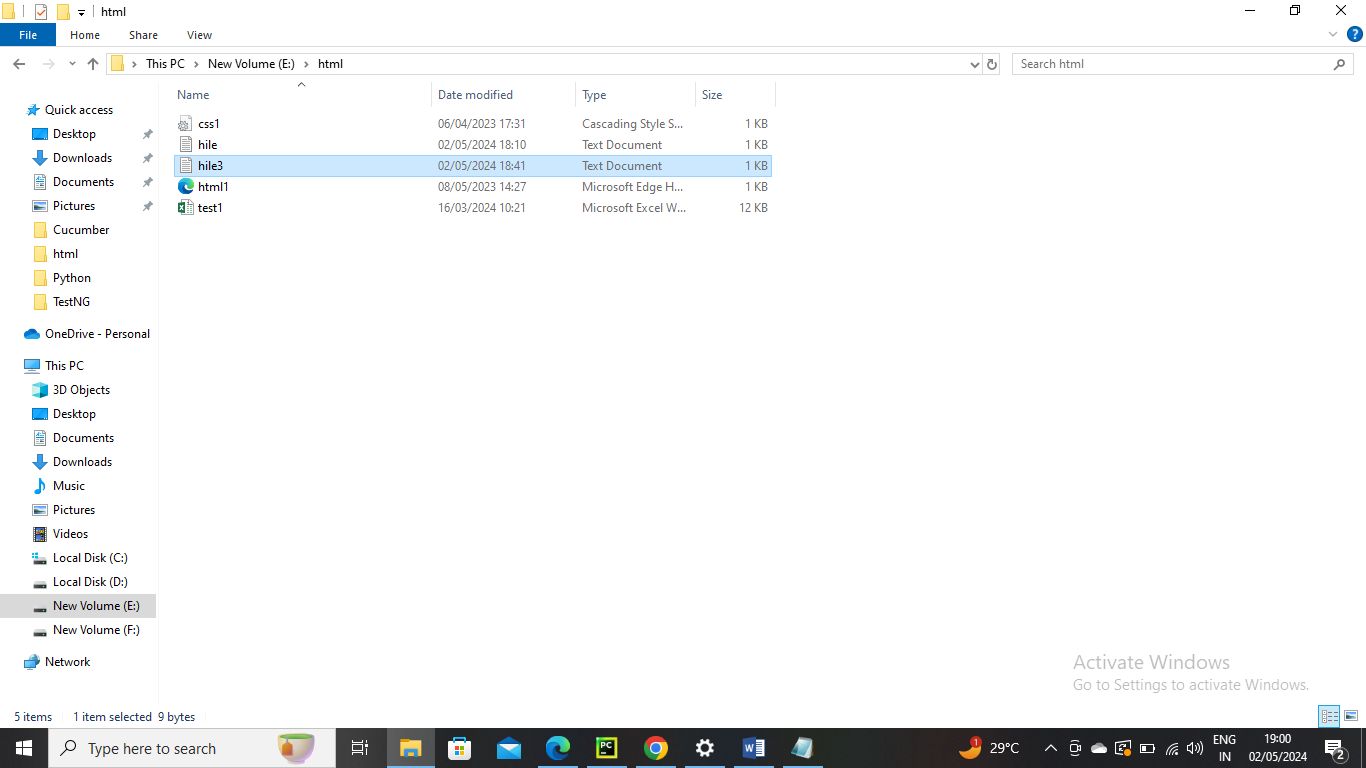


After –

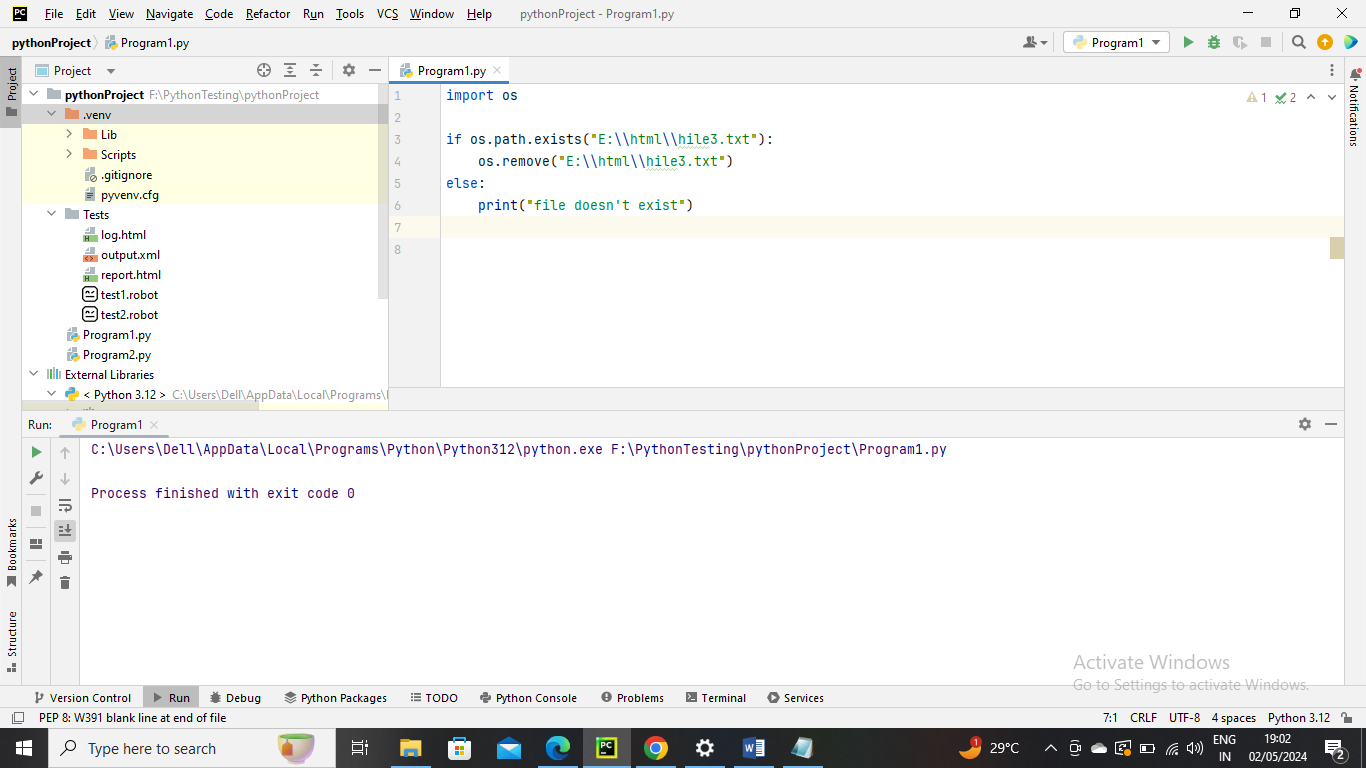


Removing the file

Before –



Program run



After –

