

Program # 6

FILE CONVERSION – REMOVE CONTROL BLOCKING DATA FROM
BACKUPS

MANUEL HOLGUIN

Table of Contents

Problem Description	3
PERL.....	3
PERL Program output Screenshots	5
Java.....	7
JAVA Output Screen shots	10
Python.....	12
Python program output screen shots	14

Problem Description

For this programming assignment we were tasked with a semi real world scenario in which an archived of backed up data became polluted or corrupted with noise inputs fabricated scenario. As per our instructions we were told to remove this noise from the txt file and only output the lines without the gibberish texts. However, this text was hard to sanitize considering the control blocks were not something a simple file reader can detect. To clean the data, the raw input must be detected and handled accordingly.

PERL

#Title: FileConverPerl Program

#Author: Manuel Holguin

#Date: 10/21/2022

#Description: The program reads the file and cleans out the noise data. If it sees a ^C triggers

the flag and ignores any input until it sees a ^B and then prints afterwards. The program

works by reading a line at a time and then for each line iterate through the individual chars

then we store the ordinance of the char in \$num and check with multiple if statements.

If flag is true print the character. In the end we have a cleaned data sheet with no noise.

#

use warnings;

use strict;

#variables used in the program

my @ASCII = "";

my \$filename = 'control-char.txt';

my \$flag = 1;

my \$ele;

#File Handler // opener

open(FH, '<', \$filename) or die \$!;

```

# Reading the file and printing the line read.
#While $stringLine is not -1 loop through
while(my $StringLine = <FH>){

    # This line separates the strings into individual characters and stores them into an array called @ASCII
    @ASCII = split(//,$StringLine);

    # for each element in the array converts it to an integers and checks if it is 2 or 3
    foreach $ele (@ASCII){
        my $num = ord($ele);

        # If the number is 3 flag is false do not print
        if($num == 3){
            $flag = 0;
        }

        # Elself it's true print the element of the array.
        elsif($flag){
            print "$ele";
        }

        # Elself num = 2 switch flag on again and print next element.
        elsif($num == 2){
            $flag = 1;
        }
    }
}

```

PERL Program output Screenshots

The screenshot shows a Visual Studio Code editor with a dark theme. The Explorer sidebar on the left shows a project with files: FileConvertPython.py, FileConverPerl.pl, control-char.txt, FileConverPerl.pl, FileConvert.java, and FileConvertPython.py. The FileConverPerl.pl file is open in the editor. The code in the editor is as follows:

```
1 #Title: FileConverPerl Program
2 #Author: Manuel Holguin
3 #Date: 10/21/2022
4 #Description: The program reads the file and cleans out the noise data. If it sees a ^C triggers
5 #           the flag and ignores any input until it sees a ^B and then prints afterwards. The program
6 #           works by reading a line at a time and then for each line iterate through the individual chars
7 #           then we store the ordiance of the char in $num and check with multiple if statements
```

Below the code editor, the Output window is open, showing the command to run the Perl script and its output:

```
[Running] perl "c:\Users\m2018\OneDrive\Documents\NMSU\Fall 2022\CS 471\assignments\programming6\File Conversion\FileConverPerl.pl"

This is line 1 This is line 1 This is line 1
This is line 2 This is line 2 This is line 2
This is line 3 This is line 3 This is line 3
This is line 4 This is line 4 This is line 4
This is line 5 This is line 5 This is line 5
This is line 6 This is line 6 This is line 6
This is line 7 This is line 7 This is line 7
This is line 8 This is line 8 This is line 8
This is line 9 This is line 9 This is line 9
This is line 10 This is line 10 This is line 10
This is line 11 This is line 11 This is line 11
This is line 12 This is line 12 This is line 12
This is line 13 This is line 13 This is line 13
This is line 14 This is line 14 This is line 14
This is line 15 This is line 15 This is line 15
This is line 16 This is line 16 This is line 16
This is line 17 This is line 17 This is line 17
This is line 18 This is line 18 This is line 18
This is line 19 This is line 19 This is line 19
```

EXPLORER

...

FileConvertPython.py

FileConverPerl.pl X

OPEN EDITORS

FileConvertPython...

X FileConverPerl.pl

FILE CONVERSION

control-char.txt

FileConverPerl.pl

FileConverPerl.java

FileConvertPython.py

FileConverPerl.pl

1 #Title: FileConverPerl Program

2 #Author: Manuel Holguin

3 #Date: 10/21/2022

4 #Description: The program reads the file and cleans out the noise data. If it sees a ^C trigger

5 # the flag and ignores any input until it sees a ^B and then prints afterwards. The pro

6 # works by reading a line at a time and then for each line iterate through the individua

7 # then we store the ordinance of the char in \$num and check with multiple if statements.

8 # If flag is true print the character. In the end we have a cleaned data sheet with no r

9 #

10

11 use warnings;

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

JUPYTER

This is line 20 This is line 20 This is line 20

This is line 21 This is line 21 This is line 21

This is line 22 This is line 22 This is line 22

This is line 23 This is line 23 This is line 23

This is line 24 This is line 24 This is line 24

This is line 25 This is line 25 This is line 25

This is line 26 This is line 26 This is line 26

This is line 27 This is line 27 This is line 27

This is line 28 This is line 28 This is line 28

This is line 29 This is line 29 This is line 29

This is line 30 This is line 30 This is line 30

This is line 31 This is line 31 This is line 31

This is line 32 This is line 32 This is line 32

This is line 33 This is line 33 This is line 33

This is line 34 This is line 34 This is line 34

This is line 35 This is line 35 This is line 35

This is line 36 This is line 36 This is line 36

This is line 37 This is line 37 This is line 37

This is line 38 This is line 38 This is line 38

This is line 39 This is line 39 This is line 39

This is line 40 This is line 40 This is line 40

This is line 41 This is line 41 This is line 41

This is line 42 This is line 42 This is line 42

This is line 43 This is line 43 This is line 43

This is line 44 This is line 44 This is line 44

This is line 45 This is line 45 This is line 45

This is line 46 This is line 46 This is line 46

This is line 47 This is line 47 This is line 47

This is line 48 This is line 48 This is line 48

This is line 49 This is line 49 This is line 49

This is line 50 This is line 50 This is line 50

> TIMELINE

Java

/*Author: Manuel Holguin

* Title: Program 6 FileConverter.java

* Date: 10/20/2022

* Description: The program takes a txt file as an input and removes

* all junk data between ^C and ^B then outputs a cleaned version of that file.

* The program does this by utilizing the java BufferedReader reading each

* individual character and storing it as an Integer in an array. In order to

* clean the data in between the control blocks a boolean called flag was used

* Whenever a ^C was located the flag would be turned off and the reader would

* continue. Otherwise if a ^B was found the flag would be activated allowing

* the character found to be stored in the array this was implemented using a while

* loop. After the file was read it is then close and finally the Integer array

* is iterated through and converted back into characters which are printed out to the

* console.

*/

import java.io.BufferedReader;

import java.io.FileNotFoundException;

import java.io.FileReader;

import java.io.IOException;

import java.util.ArrayList;

public class FileConvert {

public static void main(String[] args) throws IOException {

try {

 //This array list stores the integer values of the characters

 ArrayList<Integer> chars = new ArrayList<>();

 //open file


```

FileReader myfile = new FileReader("control-char.txt");

//BufferedReader will iterate thru the file storing the raw bits
BufferedReader br = new BufferedReader(myfile);

// yes/no flag
boolean flag = true;

//stores the bit character
int ch;

//While it is not the end of the file.
while ((ch = br.read()) != -1) {

    //If ascii = STX then flag is true
    if (ch == 2) {
        flag = true;
    }

    //else if ascii = ETX switch flag off do not store anything
    else if (ch == 3) {
        flag = false;
    }

    //Add the ch to the arraylist
    else if (flag == true) {
        chars.add(ch);
    }

    //Idk I just added to avoid any issues else just continue along soldier
    else{
        continue;
    }
}

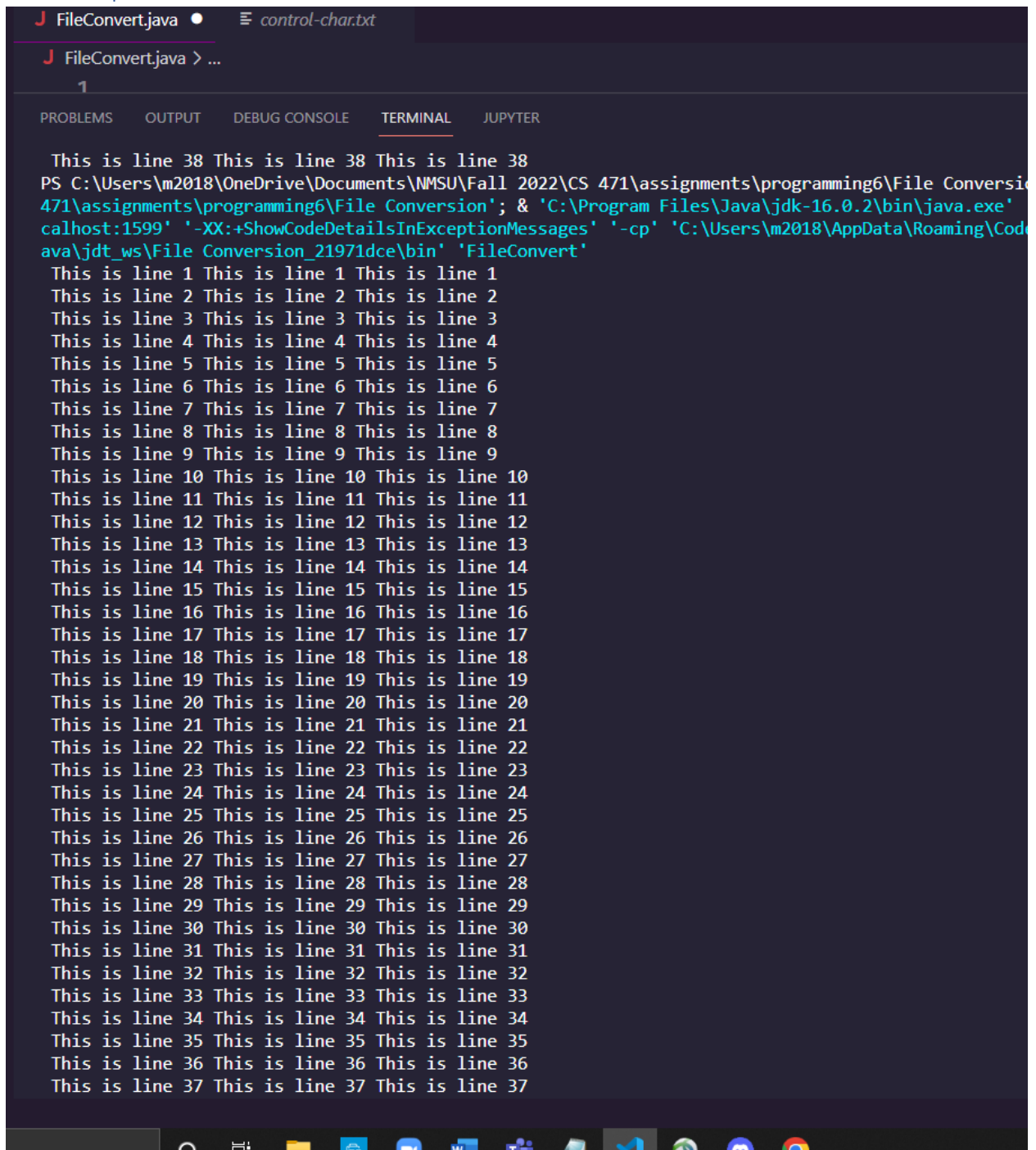
myfile.close();

//Iterate through the array and print the characters

```

```
for(int i = 0 ; i < chars.size(); i++){  
    //convert from Integer to int  
    int temp = chars.get(i);  
    //cast int to character in order to print to console.  
    char c = (char) temp;  
    System.out.print(c);  
}  
} catch (FileNotFoundException e) {  
    System.out.println("File Not Found you Dummy!!");  
    e.printStackTrace();  
}  
}  
}
```

JAVA Output Screen shots



```
FileConvert.java • control-char.txt
FileConvert.java > ...
1

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

This is line 38 This is line 38 This is line 38
PS C:\Users\m2018\OneDrive\Documents\NMSU\Fall 2022\CS 471\assignments\programming6\File Conversion_21971dce\bin' 'FileConvert'
PS C:\Users\m2018\OneDrive\Documents\NMSU\Fall 2022\CS 471\assignments\programming6\File Conversion_21971dce\bin' 'FileConvert'
This is line 1 This is line 1 This is line 1
This is line 2 This is line 2 This is line 2
This is line 3 This is line 3 This is line 3
This is line 4 This is line 4 This is line 4
This is line 5 This is line 5 This is line 5
This is line 6 This is line 6 This is line 6
This is line 7 This is line 7 This is line 7
This is line 8 This is line 8 This is line 8
This is line 9 This is line 9 This is line 9
This is line 10 This is line 10 This is line 10
This is line 11 This is line 11 This is line 11
This is line 12 This is line 12 This is line 12
This is line 13 This is line 13 This is line 13
This is line 14 This is line 14 This is line 14
This is line 15 This is line 15 This is line 15
This is line 16 This is line 16 This is line 16
This is line 17 This is line 17 This is line 17
This is line 18 This is line 18 This is line 18
This is line 19 This is line 19 This is line 19
This is line 20 This is line 20 This is line 20
This is line 21 This is line 21 This is line 21
This is line 22 This is line 22 This is line 22
This is line 23 This is line 23 This is line 23
This is line 24 This is line 24 This is line 24
This is line 25 This is line 25 This is line 25
This is line 26 This is line 26 This is line 26
This is line 27 This is line 27 This is line 27
This is line 28 This is line 28 This is line 28
This is line 29 This is line 29 This is line 29
This is line 30 This is line 30 This is line 30
This is line 31 This is line 31 This is line 31
This is line 32 This is line 32 This is line 32
This is line 33 This is line 33 This is line 33
This is line 34 This is line 34 This is line 34
This is line 35 This is line 35 This is line 35
This is line 36 This is line 36 This is line 36
This is line 37 This is line 37 This is line 37
```

J FileConvert.java • control-char.txt

J FileConvert.java > ...

1

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

```
This is line 14 This is line 14 This is line 14
This is line 15 This is line 15 This is line 15
This is line 16 This is line 16 This is line 16
This is line 17 This is line 17 This is line 17
This is line 18 This is line 18 This is line 18
This is line 19 This is line 19 This is line 19
This is line 20 This is line 20 This is line 20
This is line 21 This is line 21 This is line 21
This is line 22 This is line 22 This is line 22
This is line 23 This is line 23 This is line 23
This is line 24 This is line 24 This is line 24
This is line 25 This is line 25 This is line 25
This is line 26 This is line 26 This is line 26
This is line 27 This is line 27 This is line 27
This is line 28 This is line 28 This is line 28
This is line 29 This is line 29 This is line 29
This is line 30 This is line 30 This is line 30
This is line 31 This is line 31 This is line 31
This is line 32 This is line 32 This is line 32
This is line 33 This is line 33 This is line 33
This is line 34 This is line 34 This is line 34
This is line 35 This is line 35 This is line 35
This is line 36 This is line 36 This is line 36
This is line 37 This is line 37 This is line 37
This is line 38 This is line 38 This is line 38
This is line 39 This is line 39 This is line 39
This is line 40 This is line 40 This is line 40
This is line 41 This is line 41 This is line 41
This is line 42 This is line 42 This is line 42
This is line 43 This is line 43 This is line 43
This is line 44 This is line 44 This is line 44
This is line 45 This is line 45 This is line 45
This is line 46 This is line 46 This is line 46
This is line 47 This is line 47 This is line 47
This is line 48 This is line 48 This is line 48
This is line 49 This is line 49 This is line 49
This is line 50 This is line 50 This is line 50
```

PS C:\Users\m2018\OneDrive\Documents\NMSU\Fall 2022\CS 471\assignments\programming6\File Conversion>

Python

#Title: FileConvertPython.py

#Author: Manuel Holguin

#Date: 10-22-2022

#Description: Takes a file as input which has unneeded information inside.

```
# The program's job is to clean that unneeded information and print
# a clean version of that file. This program follows the same concept
# as the other programs uses a nested loop and checks for the ordinance
# of the character if it is either ETX or STX then this triggers the
# flag on and off and prints accordingly cleaning the data.
```

#Open the file control-char.txt

```
f = open("control-char.txt")
```

#flag boolean

```
flag = True
```

#This variable will store individual char in a string

```
fileOutput = ""
```

Now that the file is open we will begin reading the lines.

```
with open("control-char.txt") as fileobject:
```

```
    for line in fileobject:
```

```
        for character in line:
```

```
            #If statement if the ordinance of the character is 3 for (ETX) flag is false.
```

```
            if ord(character) == 3:
```

```
                flag = False
```

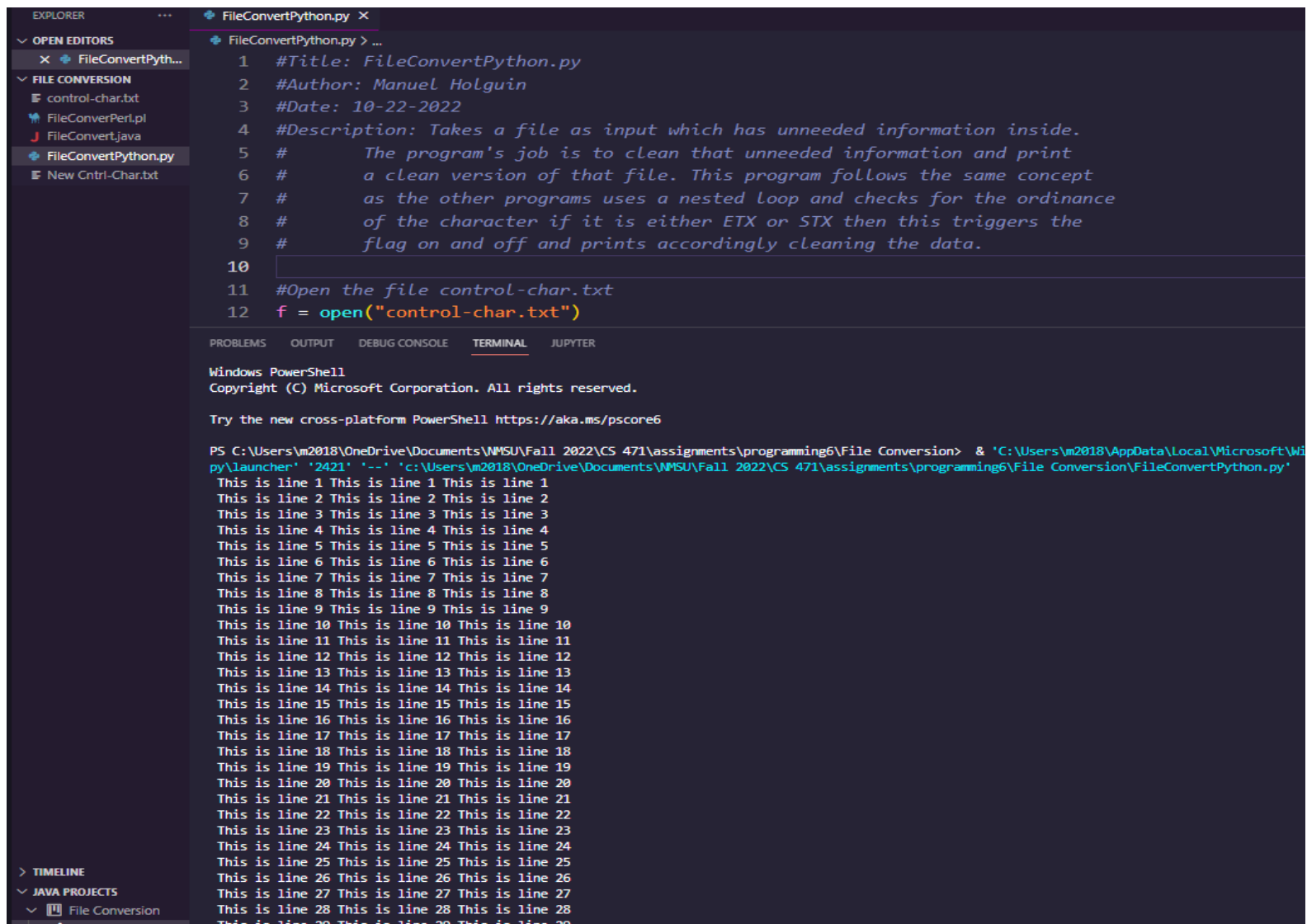
```
            # If flag is false do not print anything we are in the no go zone.
```

```
            elif flag:
```

```
                fileOutput += character
```

```
#else if ordinance of character is 2 its a (STX) switch flag on for printing.  
elif ord(character) == 2:  
    flag = True  
f.close  
  
# print out the final output.  
print (fileOutput)  
  
#Lets extend the program and write to a new file  
f2 = open ("New Cntrl-Char.txt", "w")  
f2.write(fileOutput)  
f2.close
```

Python program output screen shots



The screenshot displays the Visual Studio Code interface. The Explorer sidebar on the left shows the project structure with files like `control-char.txt`, `FileConvertPerl.pl`, `FileConvert.java`, and `FileConvertPython.py`. The main editor window shows the code for `FileConvertPython.py`. The code includes a header with title, author, date, and description, followed by a comment explaining the program's purpose and a line opening the `control-char.txt` file.

```
1  #Title: FileConvertPython.py
2  #Author: Manuel Holguin
3  #Date: 10-22-2022
4  #Description: Takes a file as input which has unneeded information inside.
5  #
6  #   The program's job is to clean that unneeded information and print
7  #   a clean version of that file. This program follows the same concept
8  #   as the other programs uses a nested loop and checks for the ordinance
9  #   of the character if it is either ETX or STX then this triggers the
10 #   flag on and off and prints accordingly cleaning the data.
11
12 #Open the file control-char.txt
13 f = open("control-char.txt")
```

The TERMINAL tab at the bottom shows the output of the program, which consists of 29 lines of the text "This is line X" where X ranges from 1 to 29. The terminal prompt indicates the command was run from a PowerShell window in the directory `C:\Users\m2018\OneDrive\Documents\WMSU\Fall 2022\CS 471\assignments\programming6\File Conversion`.

OPEN EDITORS

FILE CONVERSION

TIMELINE

JAVA PROJECTS

control-char.txt

FileConvertPerl.pl

FileConvert.java

FileConvertPython.py

New Cntrl-Char.txt

File Conversion

FileConvert

JRE System Lib...

Referenced Lib...

FileConvertPython.py > ...

```
1  #Title: FileConvertPython.py
2  #Author: Manuel Holguin
3  #Date: 10-22-2022
4  #Description: Takes a file as input which has unneeded
5  #             The program's job is to clean that unneeded
6  #             a clean version of that file. This program
7  #             as the other programs uses a nested loop and
8  #             of the character if it is either ETX or STX
9  #             flag on and off and prints accordingly cleaned
10
11 #Open the file control-char.txt
12 f = open("control-char.txt")
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

JUPYTER

This is line 10 This is line 10 This is line 10

This is line 11 This is line 11 This is line 11

This is line 12 This is line 12 This is line 12

This is line 13 This is line 13 This is line 13

This is line 14 This is line 14 This is line 14

This is line 15 This is line 15 This is line 15

This is line 16 This is line 16 This is line 16

This is line 17 This is line 17 This is line 17

This is line 18 This is line 18 This is line 18

This is line 19 This is line 19 This is line 19

This is line 20 This is line 20 This is line 20

This is line 21 This is line 21 This is line 21

This is line 22 This is line 22 This is line 22

This is line 23 This is line 23 This is line 23

This is line 24 This is line 24 This is line 24

This is line 25 This is line 25 This is line 25

This is line 26 This is line 26 This is line 26

This is line 27 This is line 27 This is line 27

This is line 28 This is line 28 This is line 28

This is line 29 This is line 29 This is line 29

This is line 30 This is line 30 This is line 30

This is line 31 This is line 31 This is line 31

This is line 32 This is line 32 This is line 32

This is line 33 This is line 33 This is line 33

This is line 34 This is line 34 This is line 34

This is line 35 This is line 35 This is line 35

This is line 36 This is line 36 This is line 36

This is line 37 This is line 37 This is line 37

This is line 38 This is line 38 This is line 38

This is line 39 This is line 39 This is line 39

This is line 40 This is line 40 This is line 40

This is line 41 This is line 41 This is line 41

This is line 42 This is line 42 This is line 42

This is line 43 This is line 43 This is line 43

This is line 44 This is line 44 This is line 44

This is line 45 This is line 45 This is line 45

This is line 46 This is line 46 This is line 46

This is line 47 This is line 47 This is line 47

This is line 48 This is line 48 This is line 48

This is line 49 This is line 49 This is line 49

This is line 50 This is line 50 This is line 50

D5-G:\Users\m2018\OneDrive\Documents\NMU\Fall-2022\CS-471\assignments\programming