

Name: M.Hassaan Atif

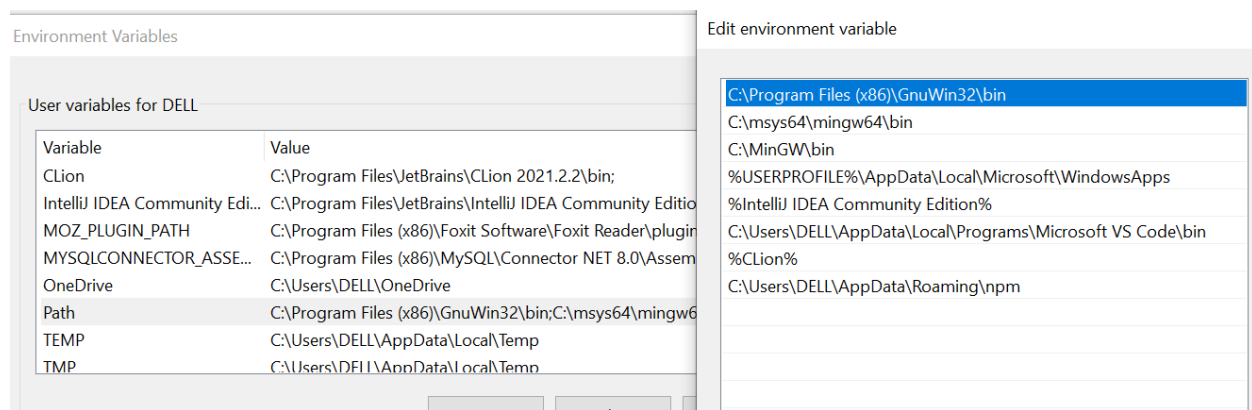
Roll No: 0147-BSCS-19

PART 1

Steps for installing Flex on my Windows 10:

1. Downloading “**GnuWin32**” from the SourceForge website (Link: <https://sourceforge.net/projects/getgnuwin32/>).
2. Downloading “**flex-2.5.4a-1.exe**” from the SourceForge website (Link: <https://sourceforge.net/projects/gnuwin32/files/flex/2.5.4a-1/>).
3. At this point, we will find Lex to be inside the bin folder of GnuWin32 program (i.e *C:\Program Files (x86)\GnuWin32\bin*).
4. Adding flex directory to “**Path**” environment variable in windows.

Path variable modification:



Root Directory of Lex:

› This PC › Local Disk (C:) › Program Files (x86) › GnuWin32 › bin

<input type="checkbox"/> Name	Date modified	Type	Size
flex	4/8/2004 2:56 AM	Application	167 KB
flex++	4/8/2004 2:56 AM	Application	167 KB
libiconv2.dll	3/15/2008 3:21 AM	Application extension	985 KB
libintl3.dll	5/7/2005 12:52 AM	Application extension	101 KB
regex2.dll	10/24/2007 3:10 PM	Application extension	78 KB
sed	12/27/2010 10:10 PM	Application	76 KB

PART 2

Example Code:

C:\Windows\System32\cmd.exe - a

```
-----  
main  
Size: 4  
Line no: 0  
Position no: 4  
Syntactic Category: KEYWORD  
  
Token ID: 55  
-----  
(  
Size: 1  
Line no: 0  
Position no: 1  
Syntactic Category: OpeningBracket  
  
Token ID: 93  
-----  
)  
Size: 1  
Line no: 0  
Position no: 2  
Syntactic Category: ClosingBracket  
  
Token ID: 92  
-----  
{  
Size: 1  
Line no: 0  
Position no: 1  
Syntactic Category: OpeningCurlyBracket  
  
Token ID: 93  
-----  
int  
Size: 3  
Line no: 1  
Position no: 3  
Syntactic Category: KEYWORD  
  
Token ID: 2  
-----
```

```
C:\Windows\System32\cmd.exe - a

C:\Users\DELL\Documents\Flex Practice>a

-----Printing Manual-----
This language is like a microsyntax of C
You may use any of the commands below:

Command: print
Usage: Prints out the details all the tokens recognized so far in the input.

Command: cls
Usage: Clears the console window

Command: exit
Usage: exits the program

Note, you may type your program here, in the console and when you want to tokenize your entire program, simply type the print command.
-----
Enter Your Program in the Console (in C format):
void main () {
    int a, b = 5;
    for (int i = 0; i<5; i = i + 1) {
        a = b + i;
    }
    float fl = 5.47;
}

print
```

Example Code 2:

```
C:\Windows\System32\cmd.exe - a

C:\Users\DELL\Documents\Flex Practice>a

-----Printing Manual-----
This language is like a microsyntax of C
You may use any of the commands below:

Command: print
Usage: Prints out the details all the tokens recognized so far in the input.

Command: cls
Usage: Clears the console window

Command: exit
Usage: exits the program

Command: out()
Usage: outputs the token into an output.txt fileNote, you may type your program here, in the console and when you want to tokenize your entire program, simply type the print command.
-----
Enter Your Program in the Console (in C format):
void main () {

int a, b =5;
for (int i = 0; i<5; i = i + 1) {
    a = b + i;
}
float pi = 3.17;
}

out()


```

Output Notification:

C:\Windows\System32\cmd.exe - a

Output has been produced into the file called output.txt in the current directory

Output generated in the text file:

ice

e View

Cut Copy path Paste shortcut Move to Copy to Delete Rename New folder New item Easy access

Organize New

This PC > Documents > Flex Practice

Name	Date modified
a	10/29/2021 10:31 PM
f	10/29/2021 10:31 PM
lex.yy	10/29/2021 10:31 PM
output	10/29/2021 10:33 PM
tokdefs	10/29/2021 7:04 PM

1.19 KB

output - Notepad

File Edit Format View Help

```
Token name: void TokenID: 1
Token name: main TokenID: 55
Token name: ( TokenID: 93
Token name: ) TokenID: 92
Token name: { TokenID: 93
Token name: int TokenID: 2
Token name: a TokenID: 111
Token name: , TokenID: 49
Token name: b TokenID: 111
Token name: = TokenID: 8
Token name: 5 TokenID: 2
Token name: ; TokenID: 90
Token name: for TokenID: 45
Token name: ( TokenID: 93
Token name: int TokenID: 2
Token name: i TokenID: 111
Token name: = TokenID: 8
Token name: 0 TokenID: 2
Token name: ; TokenID: 90
Token name: i TokenID: 111
Token name: < TokenID: 33
Token name: 5 TokenID: 2
Token name: ; TokenID: 90
Token name: i TokenID: 111
Token name: = TokenID: 8
Token name: i TokenID: 111
Token name: + TokenID: 97
Token name: 1 TokenID: 2
Token name: ) TokenID: 92
Token name: { TokenID: 93
Token name: a TokenID: 111
Token name: = TokenID: 8
Token name: b TokenID: 111
Token name: + TokenID: 97
```

Sample Code from which Tokens were generated from: void main () {

```
int a, b =5;
for (int i = 0; i<5; i = i + 1) {
    a = a + i;
}
float pi = 3.17;
}

out()
```

NOTE: The out() command is used to output to the file. If you want to display all the tokens and their info (like their line number, position, size, syntactic category) inside the terminal then simply use the print command.