## **LAB NO 4:**

Activity 1:

```
    Y Lexical Analyzer V1. Form 1

                                               ▼ 🔀 btn_Input_Click(object sender, EventArgs e)
List<string> keywordList = new List<string> { "int", "float", "while", "main", "if
int row = 1;
int count = 1;
int line_num = 0;
string[,] SymbolTable = new string[20, 6];
List<string> varListinSymbolTable = new List<string>();
ArrayList finalArray = new ArrayList();
ArrayList finalArrayc = new ArrayList();
ArrayList tempArray = new ArrayList();
char[] charinput = userInput.ToCharArray();
Regex variable_Reg = new Regex(@"^[A-Za-z_][A-Za-z0-9]*$");
Regex constants_Reg = new Regex(0^{-9}+(.[0-9]+)?([eE][+-]?[0-9]+)?");
Regex operators_Reg = new Regex(@"[-+*/><=&|]");</pre>
Regex Special_Reg = new Regex(0"[.,'\[]{}();:?]");
for (int itr = 0; itr < charinput.Length; itr++)</pre>
    string currentChar = charinput[itr].ToString();
    if (variable_Reg.IsMatch(currentChar) || constants_Reg.IsMatch(currentChar) ||
     if (variable_Reg.IsMatch(currentChar) || constants_Reg.IsMatch(currentChar)
         operators_Reg.IsMatch(currentChar) || Special_Reg.IsMatch(currentChar) ||
         charinput[itr] == ' ')
     {
         tempArray.Add(currentChar);
     if (charinput[itr] == '\n')
         if (tempArray.Count != 0)
             finalArray.Add(string.Join("", tempArray.ToArray()));
             tempArray.Clear();
if (tempArray.Count != 0)
     finalArray.Add(string.Join("", tempArray.ToArray()));
     tempArray.Clear();
```

```
foreach (string line in finalArray)
{
    line_num++;
    char[] lineChar = line.ToCharArray();

for (int itr = 0; itr < lineChar.Length; itr++)
{
    string currentChar = lineChar[itr].ToString();

    if (variable_Reg.IsMatch(currentChar) || constants_Reg.IsMatch(currentChar)
{
        tempArray.Add(currentChar);
}

if (lineChar[itr] == ' ' || operators_Reg.IsMatch(currentChar) || Special_
{
        if (tempArray.Count != 0)
        {
            finalArrayc.Add(string.Join("", tempArray.ToArray()));
            tempArray.Clear();
        }

        if (operators_Reg.IsMatch(currentChar) || Special_Reg.IsMatch(currentCourrentChar)</pre>
```

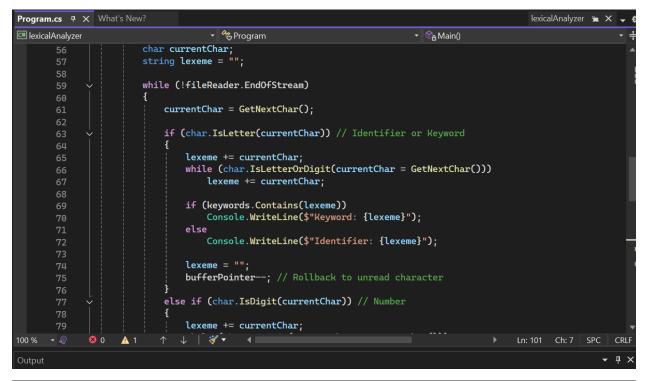
## Output:

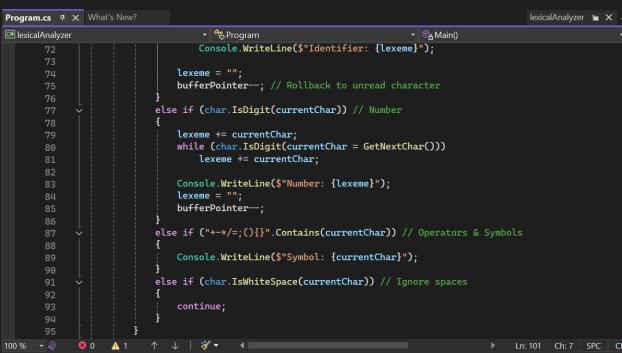
## Graded Task 1:

```
Œ lexicalAnalyzer
                                        ▼ % Program
                                                                                    ▼ 😭 Main()
              v using System;
  { ⅓
                using System.Collections.Generic; using System.IO;
              3 references

v class LexicalAnalyzer
                     const int BUFFER_SIZE = 10; // Define buffer size
char[] bufferA = new char[BUFFER_SIZE];
                     char[] bufferB = new char[BUFFER_SIZE];
                     int bufferPointer = 0;
                     bool isBufferAActive = true;
                     StreamReader fileReader;
                     HashSet<string> keywords = new HashSet<string> { "if", "else", "while", "return", "int", "floa
                     public LexicalAnalyzer(string filePath)
                          fileReader = new StreamReader(filePath);
                         LoadBuffer(); // Load initial buffer
                             • 1 ×
```

```
Program.cs ₹ X What's New?
                                                                                                                 lexicalAnalyzer 🛎 🗙 🗸 ♦ Solution Explorer
☐ lexicalAnalyzer
                                          → % Program
                                                                                       → 🏖 Main()
                                                                                                                                           ₽ 0 + 5 8
                           if (isBufferAActive)
                                Array.Clear(bufferB, 0, BUFFER_SIZE);
fileReader.Read(bufferB, 0, BUFFER_SIZE);
                                                                                                                                            Solution 'lexic
                                                                                                                                            Depend
                           else
                                                                                                                                              D C# Program
                                Array.Clear(bufferA, 0, BUFFER_SIZE); fileReader.Read(bufferA, 0, BUFFER_SIZE);
        36
37
38
39
                           isBufferAActive = !isBufferAActive; // Swap buffers
                           bufferPointer = 0;
                       private char GetNextChar()
                            if (bufferPointer >= BUFFER_SIZE)
                                                                                                                                          LoadBuffer();
                                bufferPointer = 0;
```





```
Program.cs 7 🗶 What's New?
                                                                                                     lexicalAnal
                                     ▼ % Program
IexicalAnalyzer
                                                                              ▼ 😭 Main()
                    }
               0 references
               class Program
      101 P
                    static void Main()
                        Console.Write("Enter file path to analyze: ");
                        string filePath = Console.ReadLine();
                        if (!File.Exists(filePath))
                            Console.WriteLine("Error: File not found.");
                            return;
                        LexicalAnalyzer lexer = new LexicalAnalyzer(filePath);
                        lexer.Analyze();
                        Console.WriteLine("\nLexical Analysis Completed.");
               }
     - Q
00 %
             ⊗ 0
                               ↓ | 😽 🕶
                    A 1
                                                                                                  Ln: 101
Output
```

## Output:

```
Enter file path to analyze: C:\Users\HP\Documents\test.txt

Keyword: int
Identifier: x
Symbol: =
Number: 10
Symbol: f
Symbol: (
Identifier: x
Symbol: )
Symbol: 5
Symbol: 1
Symbol: 1
Symbol: 1
Symbol: 2
Symbol: 5
Symbol: 5
Symbol: 7
Symbol: 8
Symbol: 7
Symbol: 8
Symbol: 9
Symbol: 9
Symbol: 1
Symbol: 2
Symbol: 3
Symbol: 3
Symbol: 3
Symbol: 7
Symbol: 8
Symbol: 8
Symbol: 9
Symb
```