



CSE439s: Design of Compilers

Compiler Project – Second Milestone (Parser)

Team Members

Name	Id	section
Beshoy Ashraf Faheem	1805453	3
Engy Fayek Adeeb	1806935	3
Kirollos Ashraf Sedky	1807624	3
Antonius Adel Ragheb Nan	1805824	5
Kirollos Medhat Massoud	1809488	3
Philopateer Emad Samir	1802215	3

Drive Link

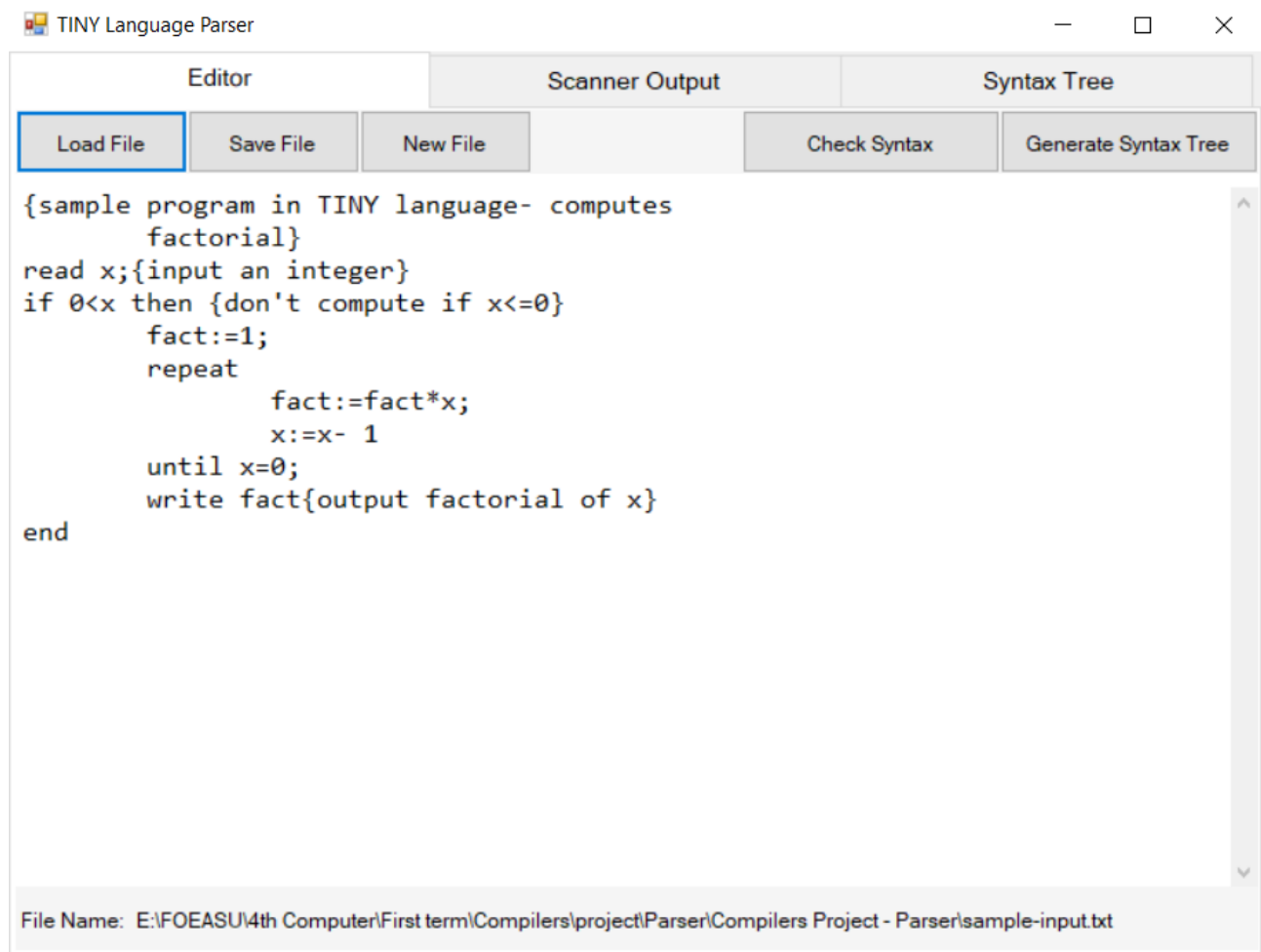
https://drive.google.com/drive/folders/1SSC6Mf54Wpk1-7PgxoA1RNOALUCHUQX3?usp=share_link

It contains:

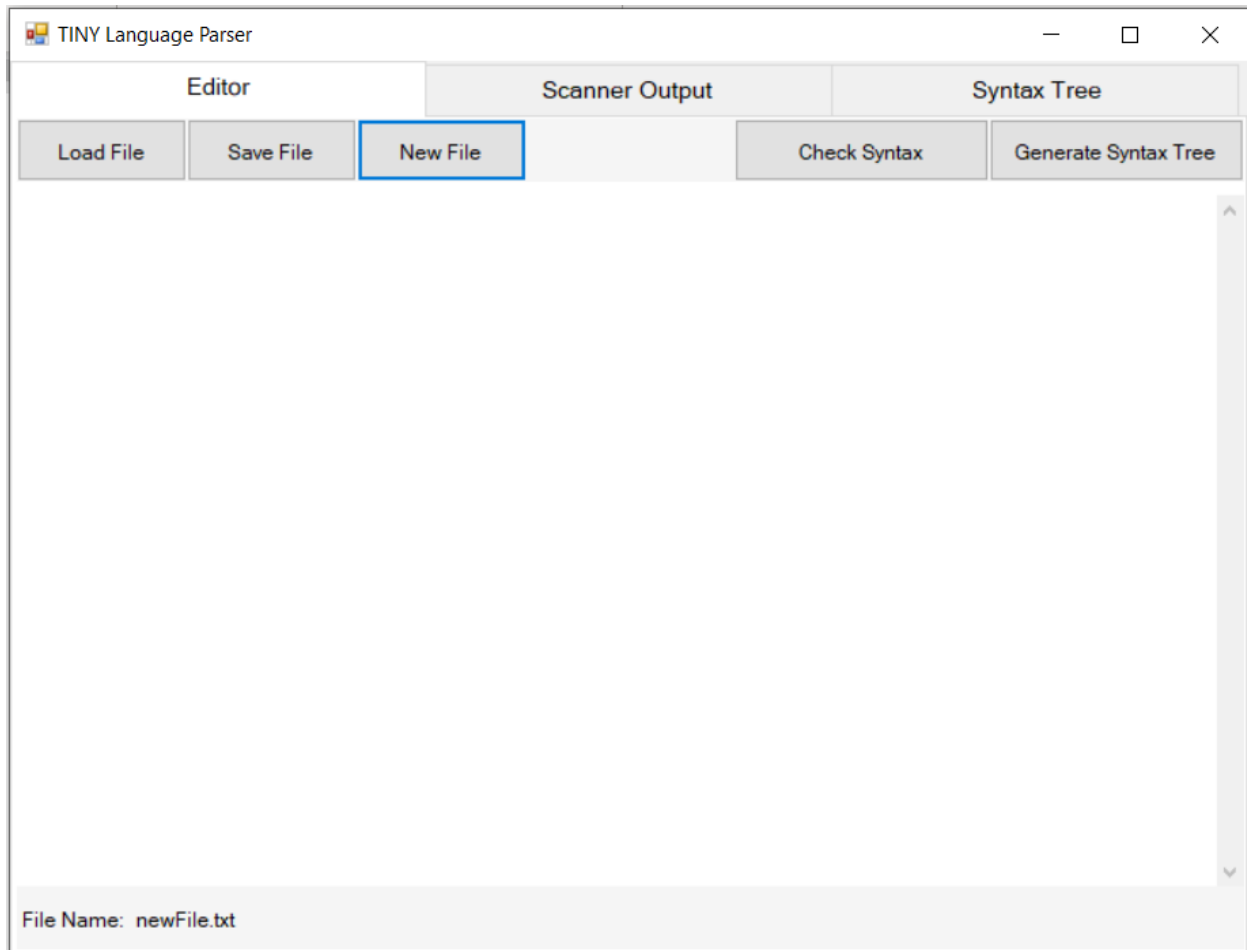
- the source code (Compilers Project - Parser.zip)
- the exe with a sample input (exe + sample-input.zip)

Features and Screenshots

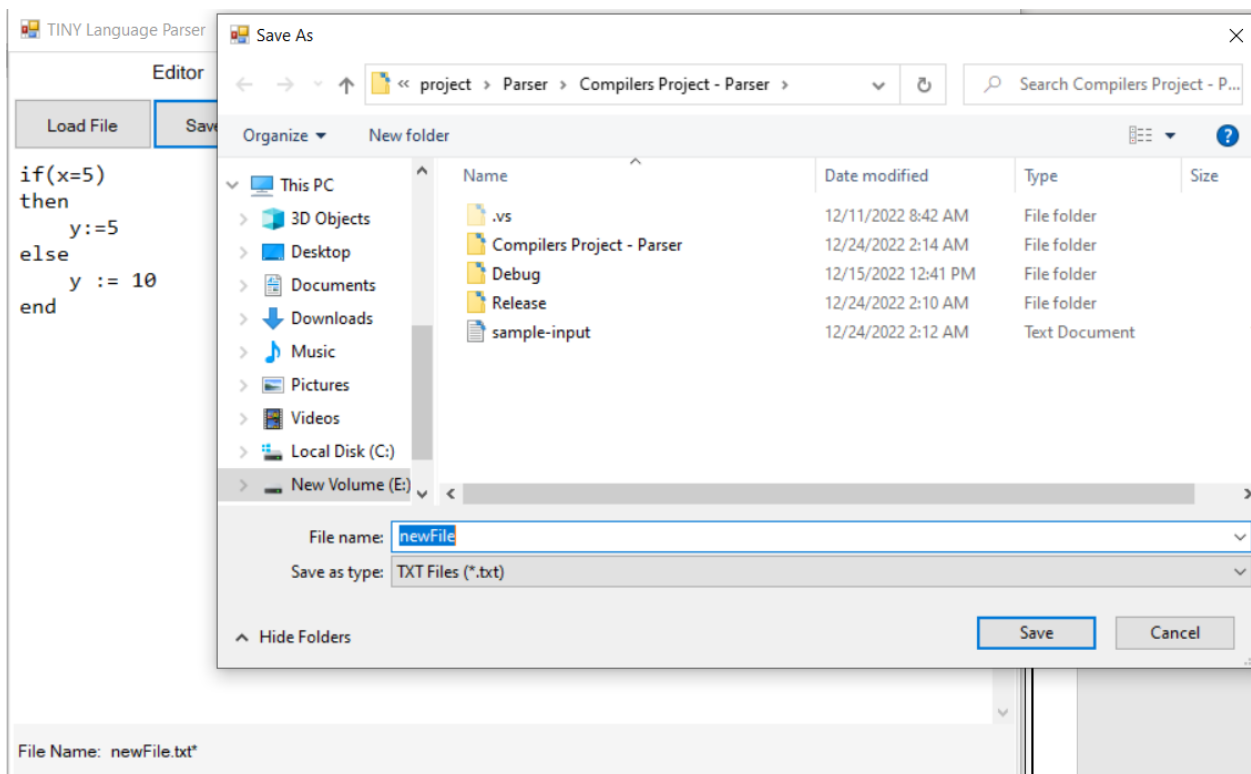
1- Editor tab with the ability to load code from a file



2- Create new file in the editor and save it later



3- Ability to save a newly created file and save later changes to it, or save changes to a loaded file

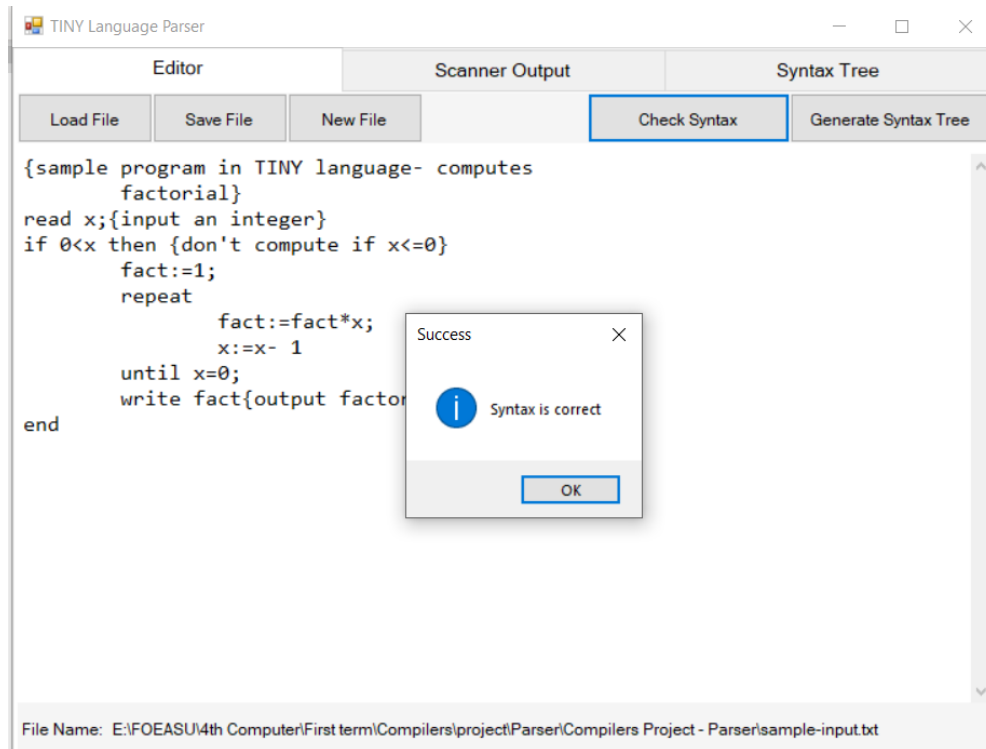


- 4- Name of opened file is displayed at the bottom bar, or displays newFile if the file is created from the editor.
also the file name at the bottom shows whether the file is saved or not by displaying an (*) next the file name if it's changed

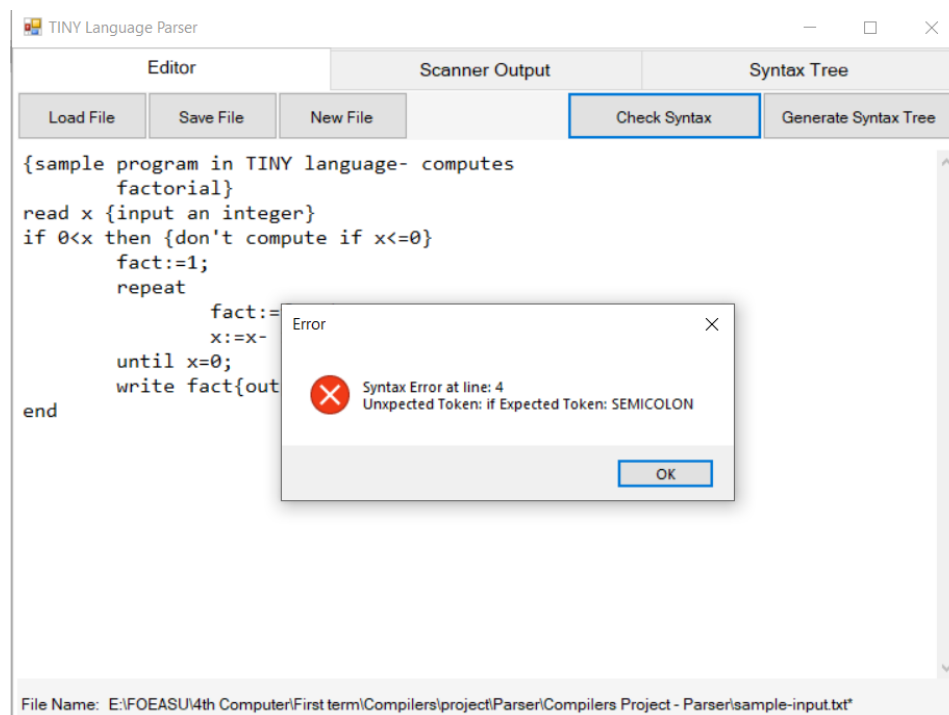
5-

File Name: E:\FOEASU\4th Computer\First term\Compilers\project\Parser\Compilers Project - Parser\sample-input.txt

- 6- Ability to check syntax and show if the syntax is correct or not



- 7- Shows a list of syntax errors with the line number and helpful messages



TINY Language Parser

Editor Scanner Output Syntax Tree

Load File Save File New File Check Syntax Generate Syntax Tree

```
{sample program in TINY language- computes
    factorial}
read x {input an integer}
if 0<x {don't compute if x<=0}
    fact:=1;
    repeat
        fact:=
        x:=x-
    until x=0;
    write fact{out
end
```

File Name: E:\FOEASU\4th Computer\First term\Compilers\project\Parser\Compilers Project - Parser\sample-input.txt*

Error

Syntax Error at line: 4
Unexpected Token: if Expected Token: SEMICOLON

Syntax Error at line: 5
Unexpected Token: fact Expected Token: THEN

OK

TINY Language Parser

Editor Scanner Output Syntax Tree

Load File Save File New File Check Syntax Generate Syntax Tree

```
{sample program in TINY language- computes
    factorial}
read x{input an integer}
if 0<x {don't compute if x<=0}
    fact=1
    repeat
        fact:=fact*x;
        x:=x- 1
    until x=0;
    write fact{output fact
end
```

File Name: E:\FOEASU\4th Computer\First term\Compilers\project\Parser\Compilers Project - Parser\sample-input.txt*

Error

Syntax Error at line: 4
Unexpected Token: if Expected Token: SEMICOLON

Syntax Error at line: 5
Unexpected Token: fact Expected Token: THEN

Syntax Error at line: 5
Unexpected Token: = Expected Token: ASSIGN

Syntax Error at line: 5
Unexpected token: =

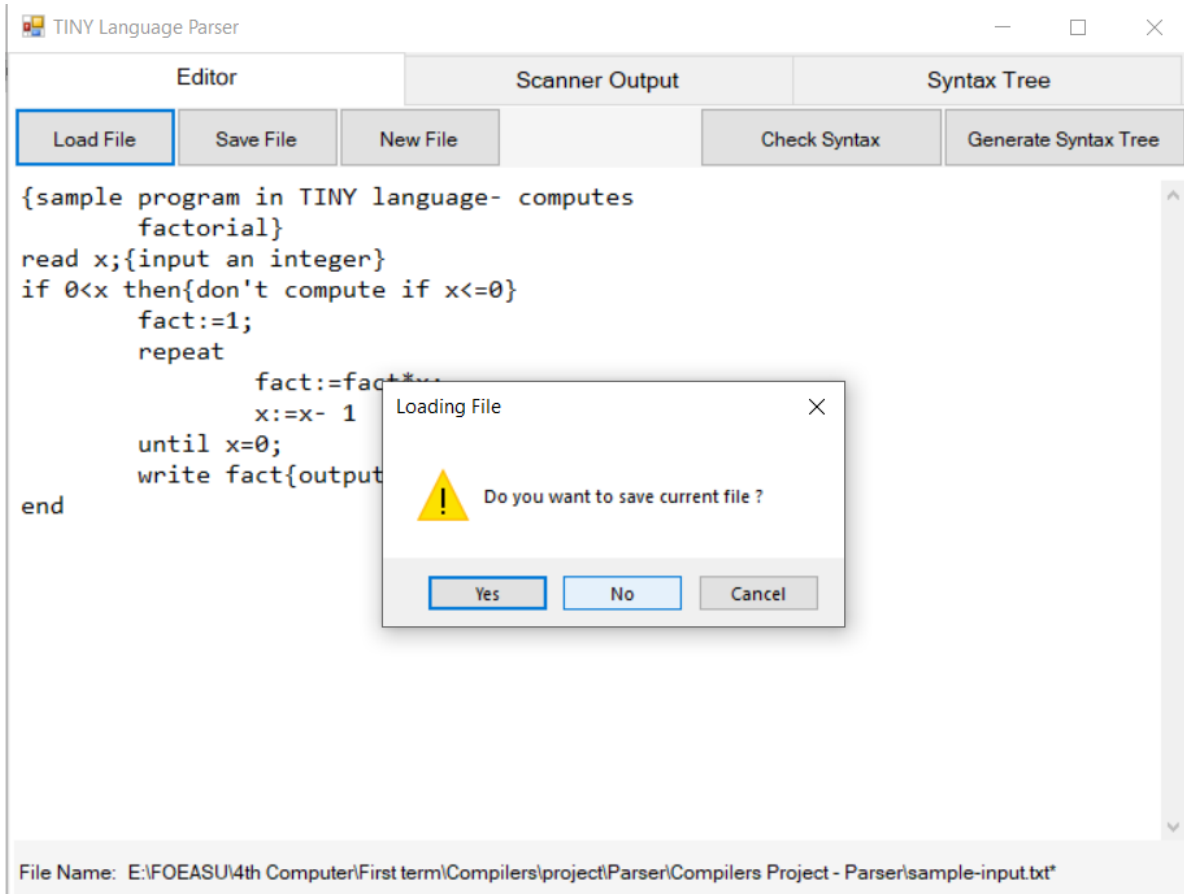
Syntax Error at line: 5
Unexpected Token: 1 Expected Token: SEMICOLON

Syntax Error at line: 5
Unexpected token: 1

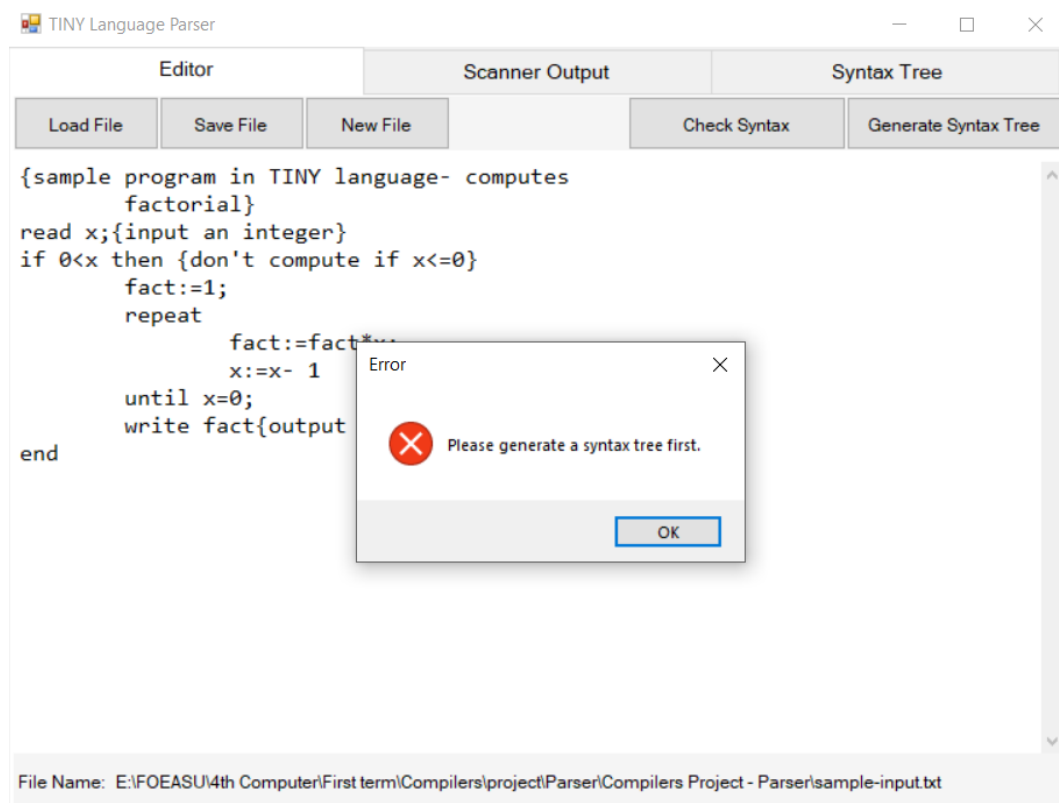
Syntax Error at line: 6
Unexpected Token: repeat Expected Token: SEMICOLON

OK

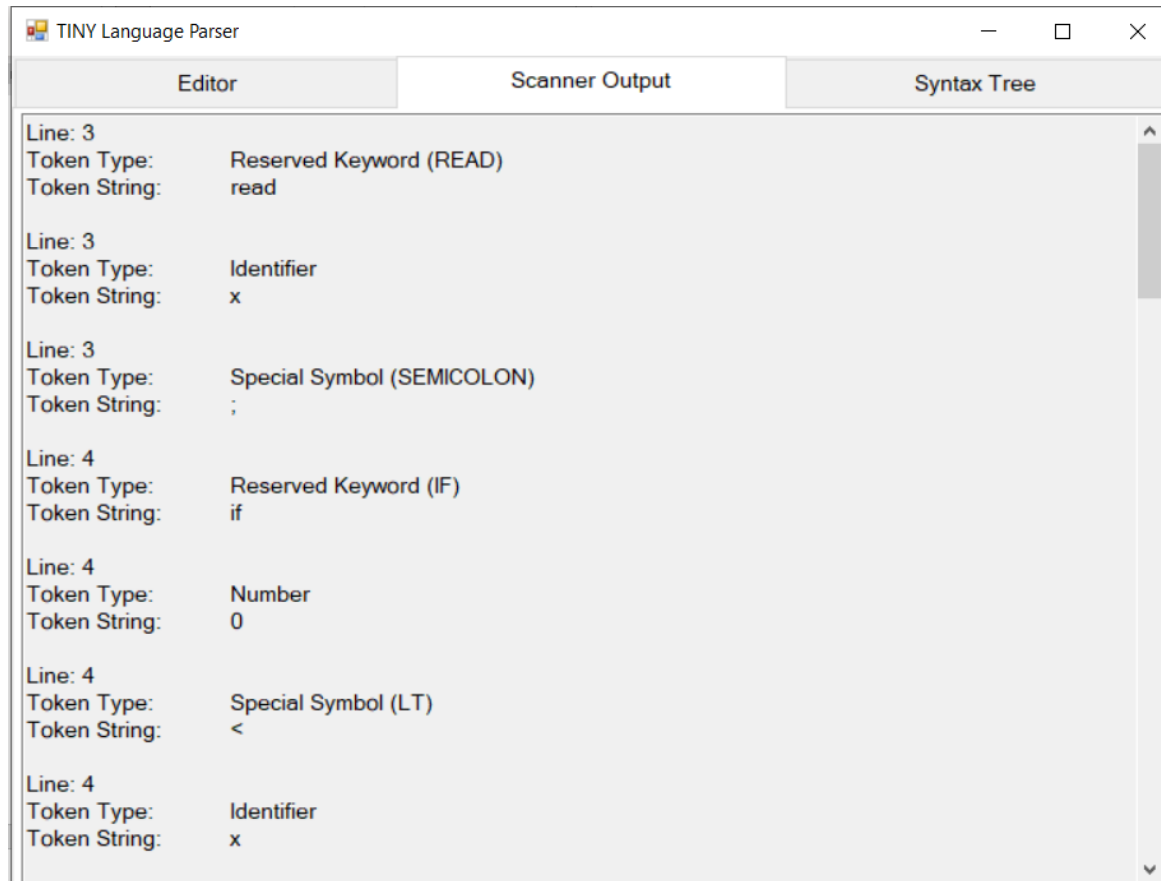
- 8- Shows an alert if the user changed the file and tried to open a new file or load one (overwrite the open file)



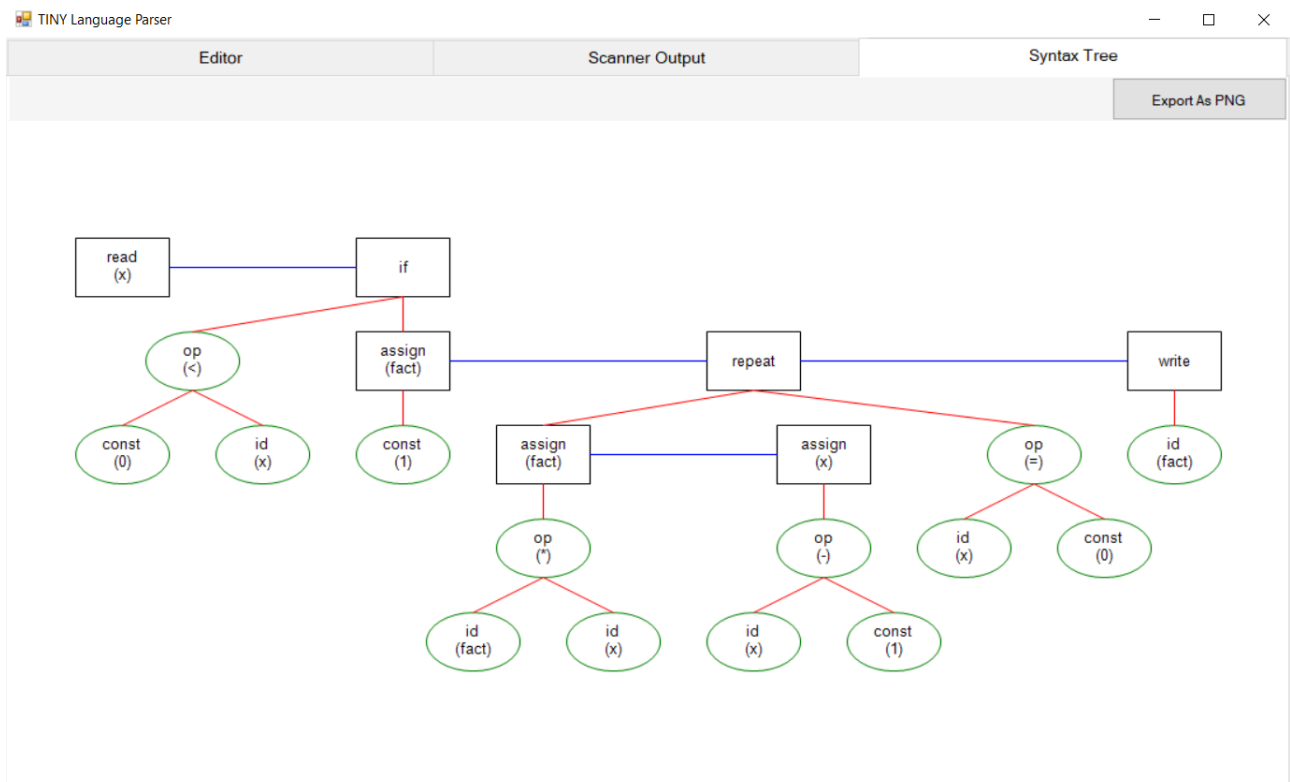
- 9- The Scanner output tab and the Syntax tree tab are disabled by default till check syntax or Generate Syntax tree are pressed respectively



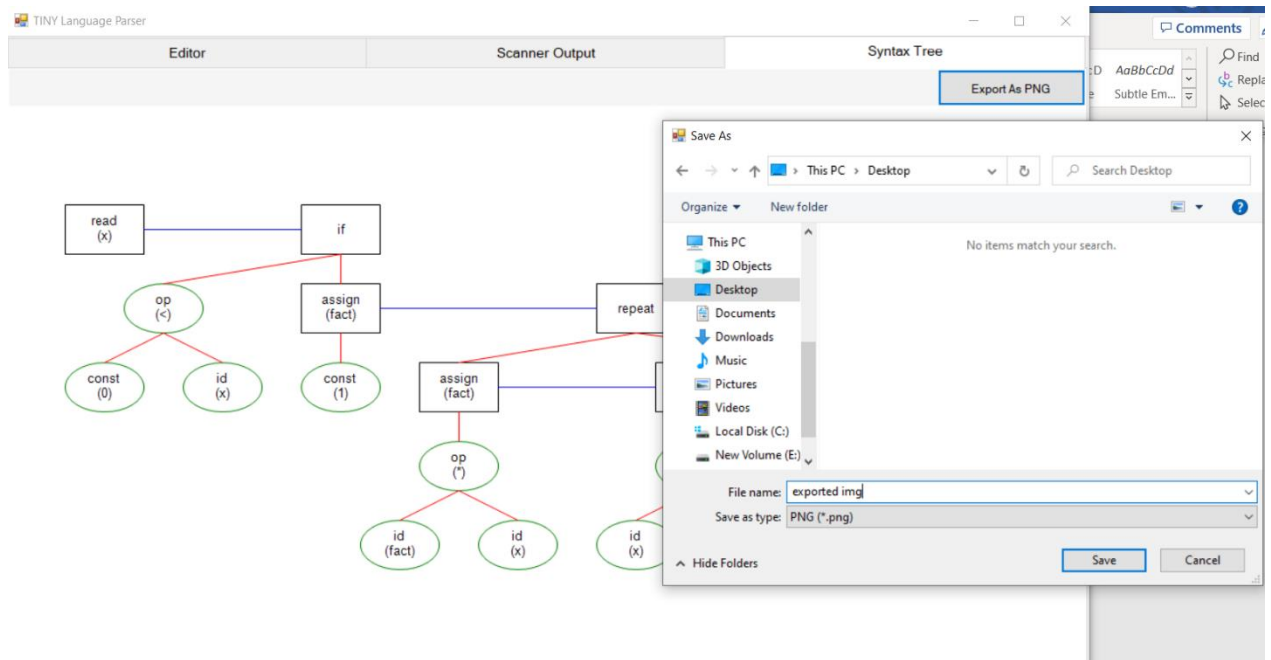
- 10- Shows the scanner output (tokens)
line number, token type, token string



- 11- Draw the syntax tree, implemented natively without the use of any external libraries, the algorithm used for drawing the tidy tree is completely dynamic and prevents overlapping of tree nodes



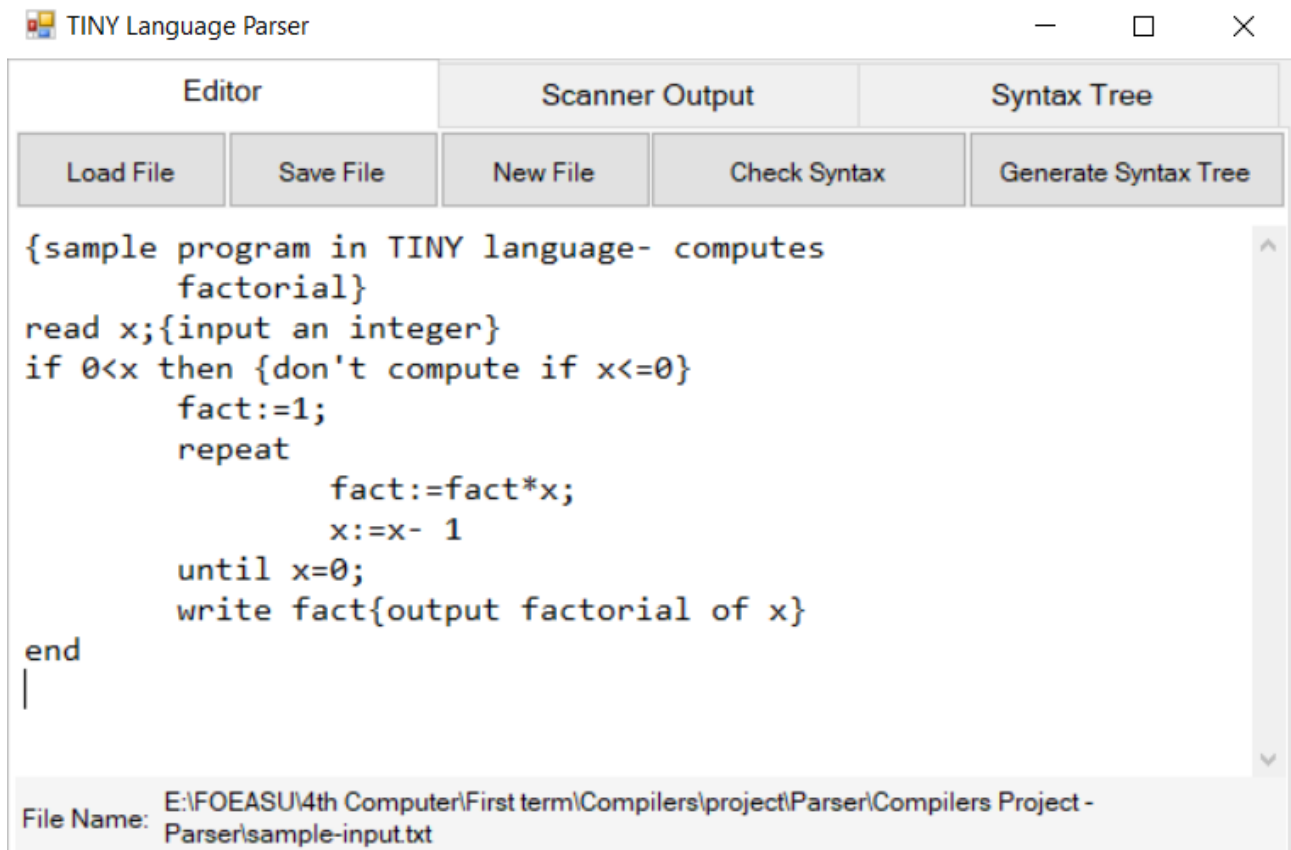
12- Allows exporting the generated syntax tree as a png of high resolution

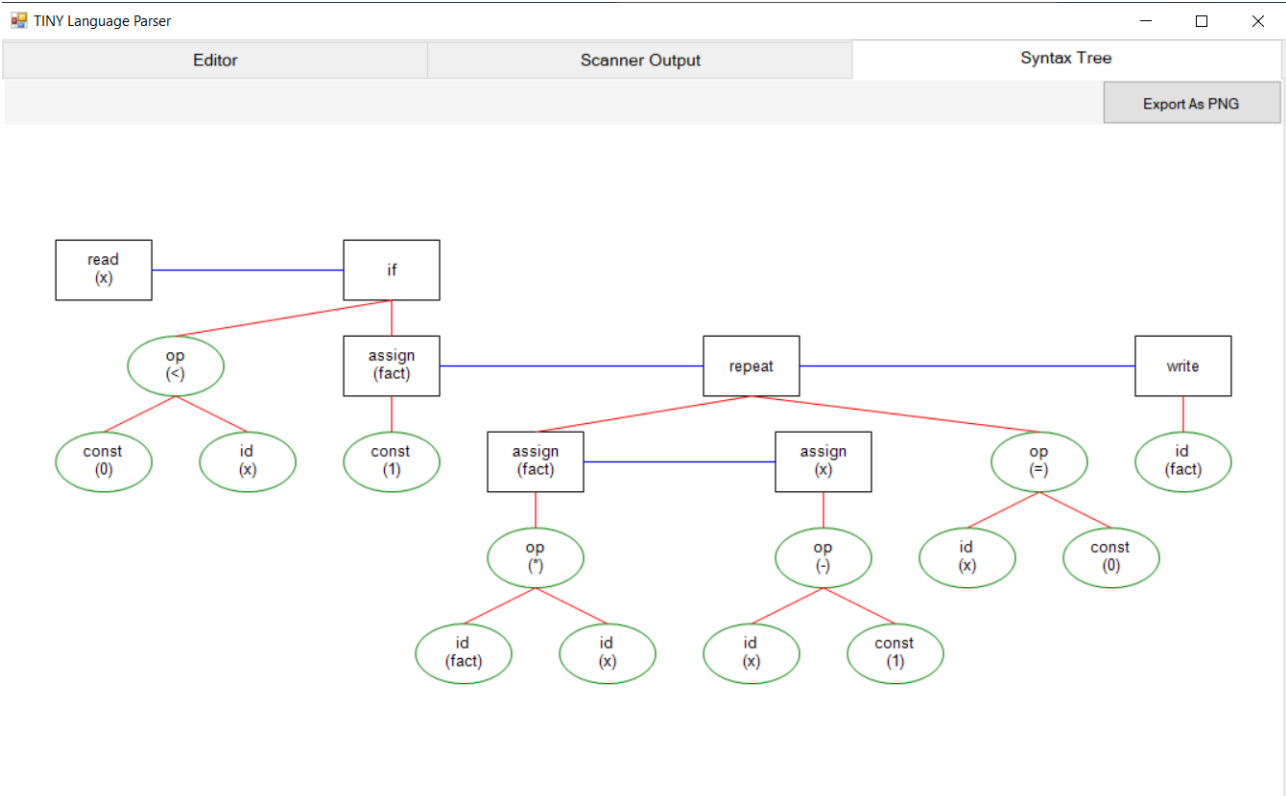


13- The GUI is completely Responsive with any screen size

Sample Programs with the corresponding syntax tree

1-





2-

TINY Language Parser

Editor Scanner Output Syntax Tree

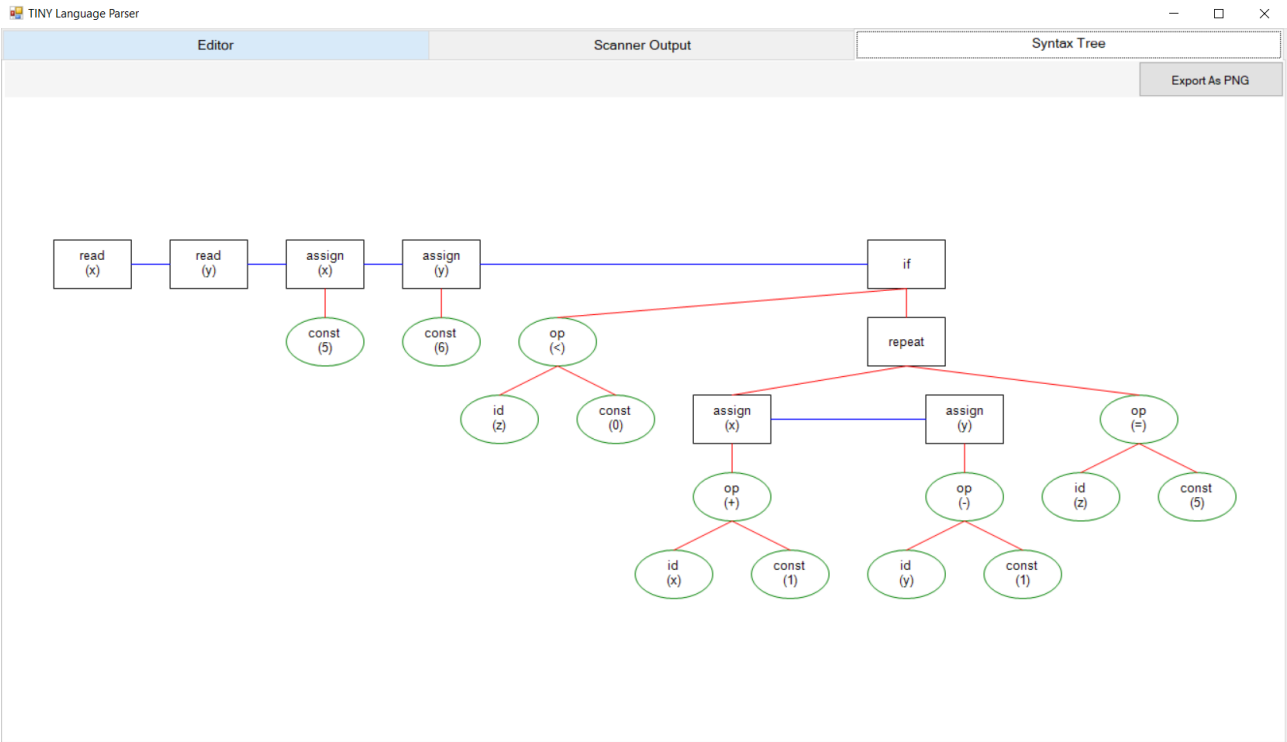
Load File Save File New File Check Syntax Generate Syntax Tree

```

read x;
read y;
x:=5;
y:=6;

if(z < 0) then
    repeat
        x := x + 1;
        y := y - 1
    until z = 5
    {this is a comment}
end
  
```

File Name: E:\FOEASU\4th Computer\First term\Compilers\project\Parser\Compilers Project - Parser\sample-input.txt*



3-

TINY Language Parser

Editor Scanner Output Syntax Tree

Load File Save File New File Check Syntax Generate Syntax Tree

```

y := 6;
if (x < 0) then
  repeat
    x := x + 1;
    y := y + 1;
    if (x < 0) then
      repeat
        x := x + 1;
        y := y + 1
      until x = 5
    end
  until x = 5
end
  
```

File Name: newFile.txt*

