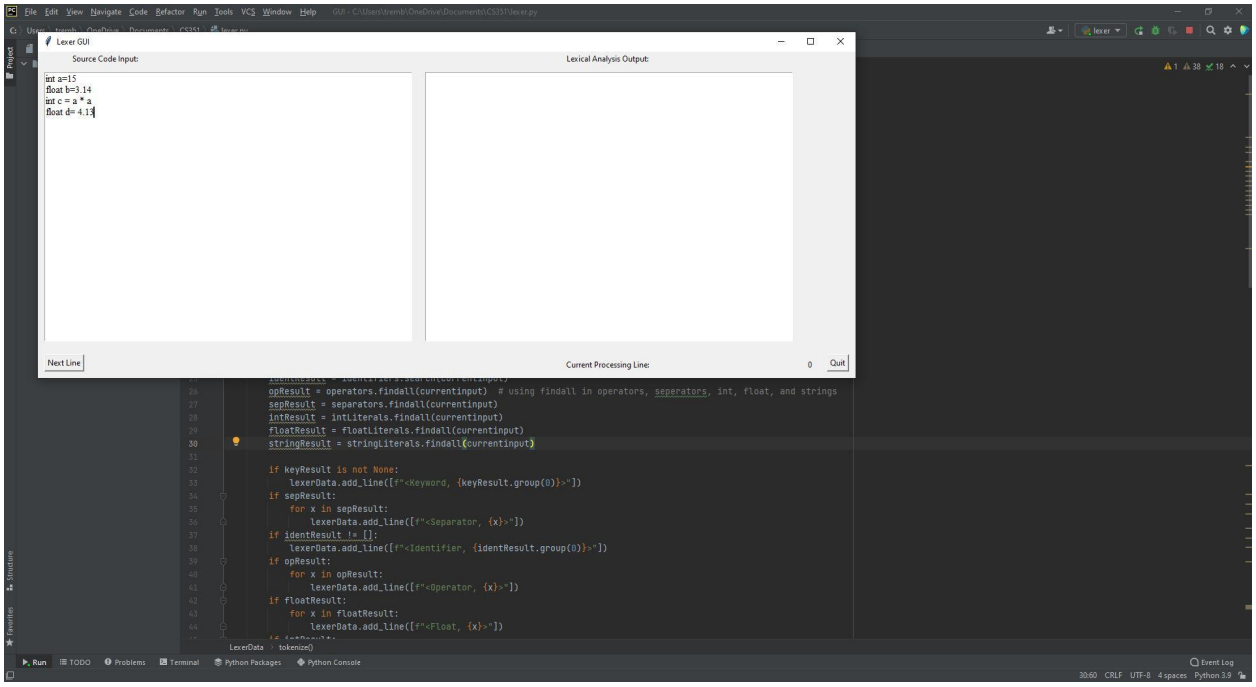
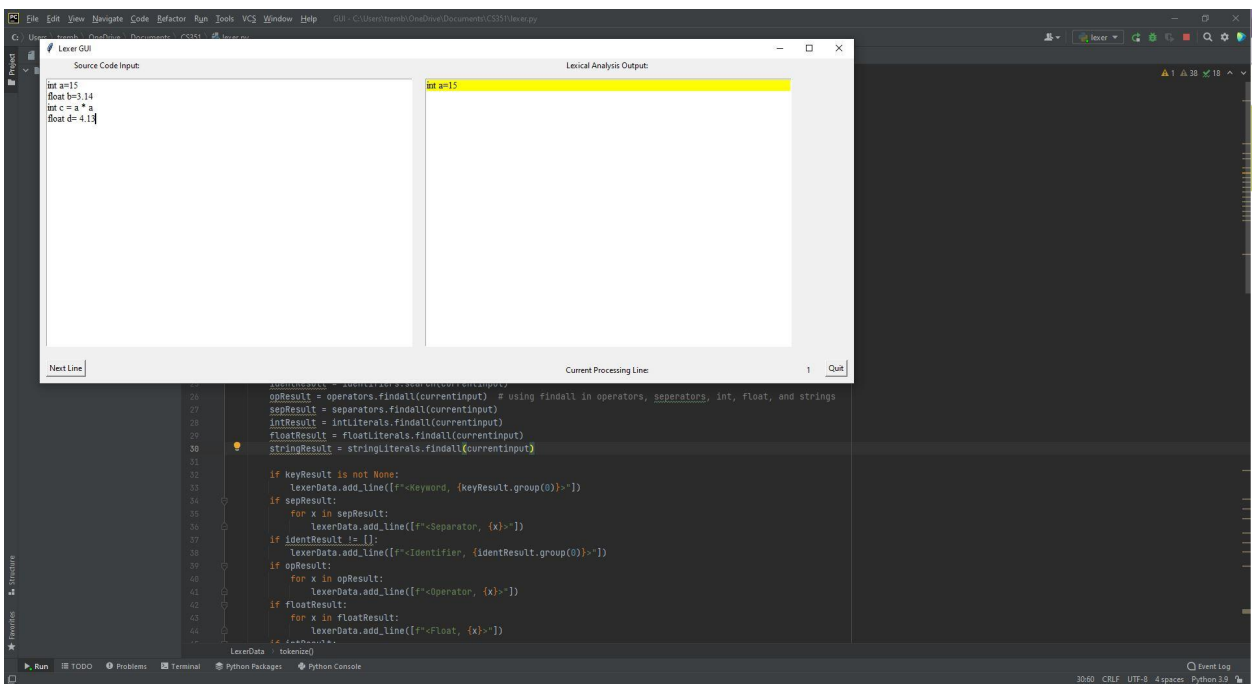
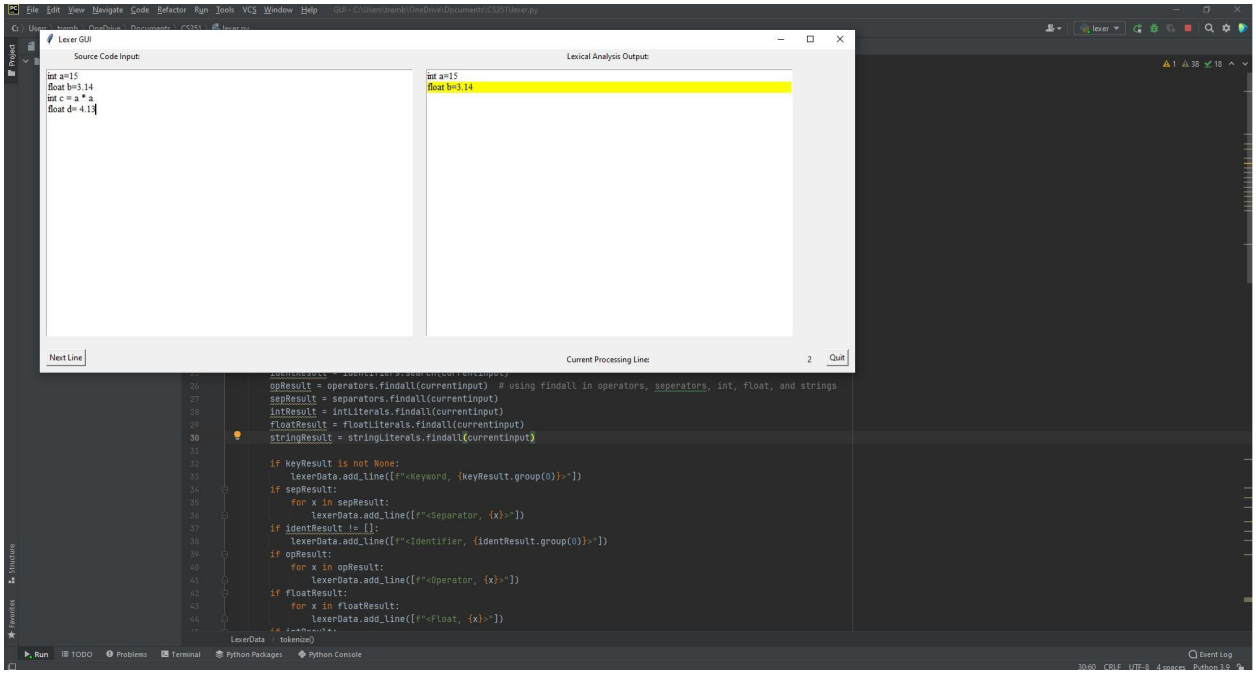
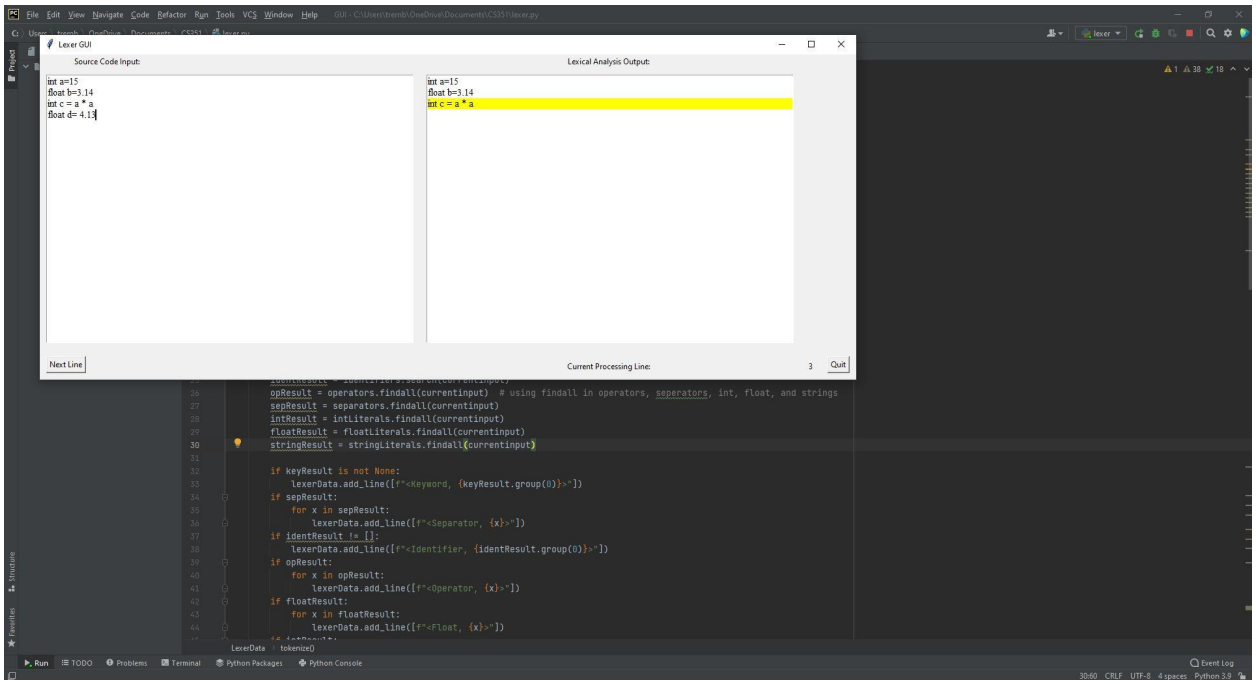
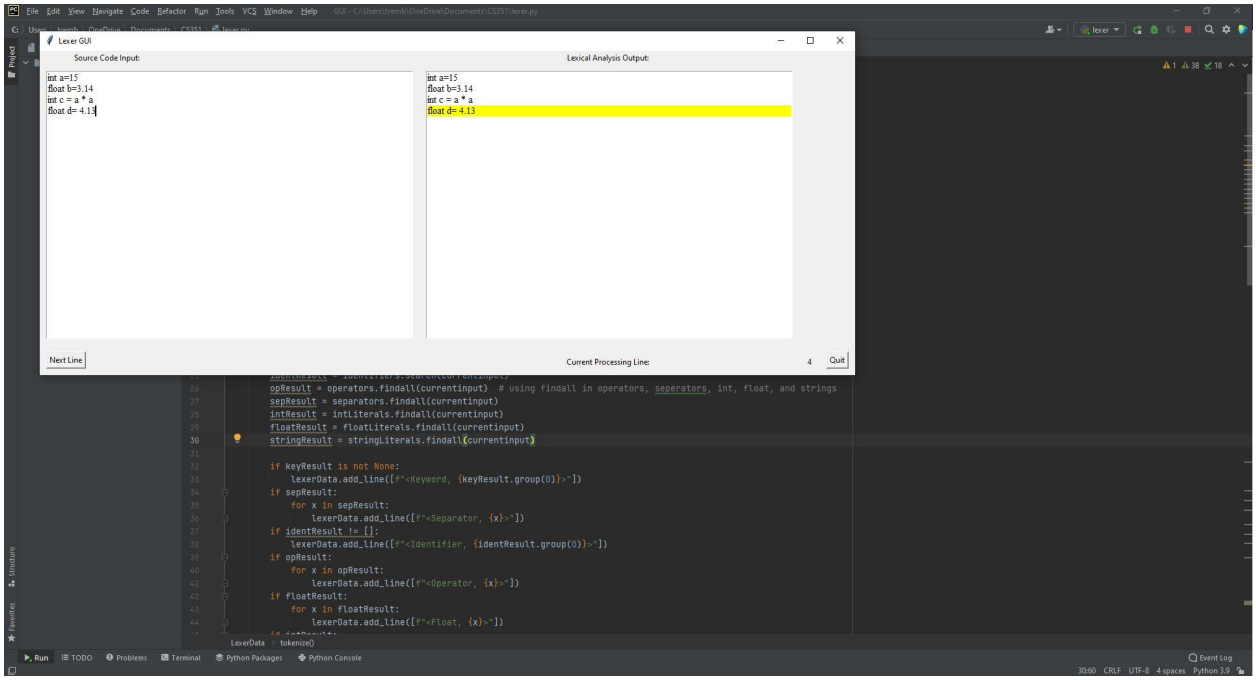
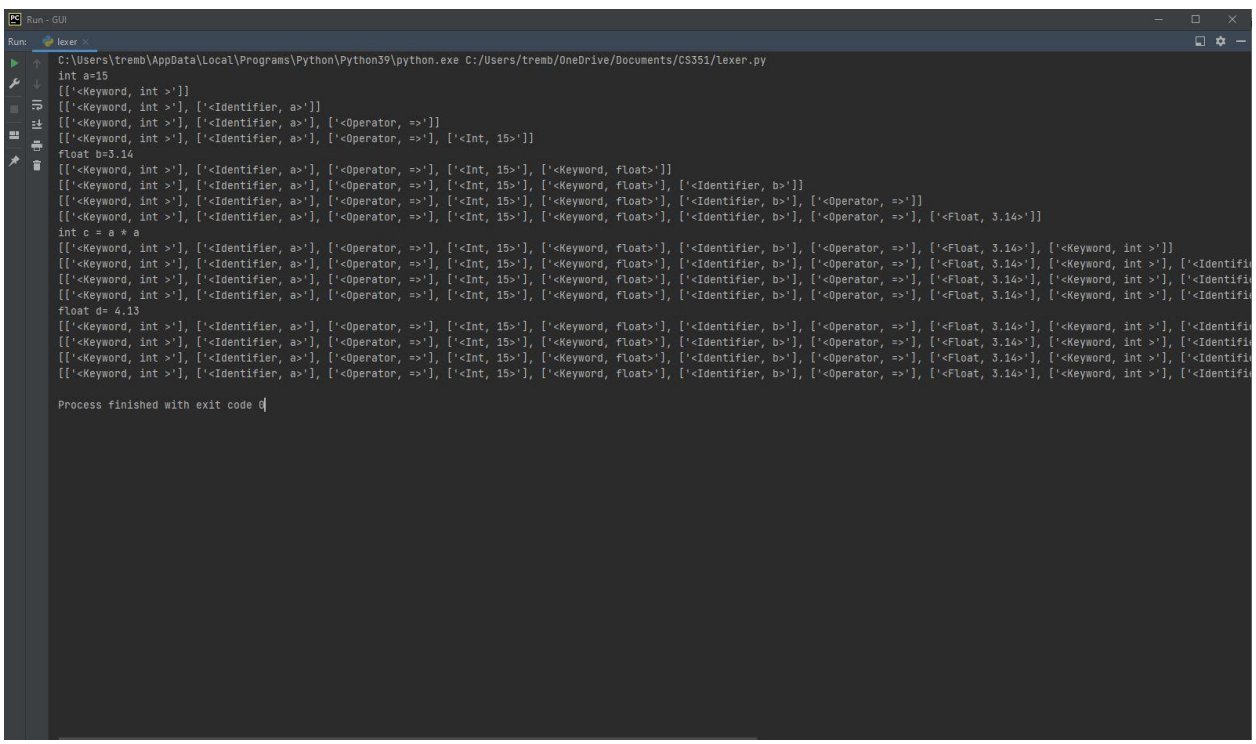

Lexer GUI
Source Code Input:
Lexical Analysis Output:
int a=15
float b=3.14
int c = a * a
float d= 4.13
Next Line
Current Processing Line:
0
Quit
opResult = operators.findall(currentinput) # using findall in operators, seperators, int, float, and strings
sepResult = separators.findall(currentinput)
intResult = intLiterals.findall(currentinput)
floatResult = floatLiterals.findall(currentinput)
stringResult = stringLiterals.findall(currentinput)
if keyResult is not None:
lexerData.add_line([f"<Keyword, {keyResult.group(0)}>"])
if sepResult:
for x in sepResult:
lexerData.add_line([f"<Separator, {x}>"])
if identResult != []:
lexerData.add_line([f"<Identifier, {identResult.group(0)}>"])
if opResult:
for x in opResult:
lexerData.add_line([f"<Operator, {x}>"])
if floatResult:
for x in floatResult:
lexerData.add_line([f"<Float, {x}>"])


Lexer GUI
Source Code Input:
Lexical Analysis Output:
int a=15
float b=3.14
int c = a * a
float d= 4.13
int a=15
Next Line
Current Processing Line:
1
Quit
opResult = operators.findall(currentinput) # using findall in operators, seperators, int, float, and strings
sepResult = separators.findall(currentinput)
intResult = intLiterals.findall(currentinput)
floatResult = floatLiterals.findall(currentinput)
stringResult = stringLiterals.findall(currentinput)
if keyResult is not None:
lexerData.add_line([f"<Keyword, {keyResult.group(0)}>"])
if sepResult:
for x in sepResult:
lexerData.add_line([f"<Separator, {x}>"])
if identResult != []:
lexerData.add_line([f"<Identifier, {identResult.group(0)}>"])
if opResult:
for x in opResult:
lexerData.add_line([f"<Operator, {x}>"])
if floatResult:
for x in floatResult:
lexerData.add_line([f"<Float, {x}>"])


Lexer GUI
Source Code Input:
int a=15
float b=3.14
int c = a * a
float d= 4.13
Lexical Analysis Output:
int a=15
float b=3.14
Next Line
Current Processing Line:
2
Quit


Lexer GUI
Source Code Input:
int a=15
float b=3.14
int c = a * a
float d= 4.13
Lexical Analysis Output:
int a=15
float b=3.14
int c = a * a
Next Line
Current Processing Line:
3
Quit


Lexer GUI
Source Code Input:
int a=15
float b=3.14
int c = a * a
float d= 4.13
Lexical Analysis Output:
int a=15
float b=3.14
int c = a * a
float d= 4.13
Next Line
Current Processing Line:
4
Quit
opResult = operators.findall(currentinput) # using findall in operators, seperators, int, float, and strings
sepResult = separators.findall(currentinput)
intResult = intLiterals.findall(currentinput)
floatResult = floatLiterals.findall(currentinput)
stringResult = stringLiterals.findall(currentinput)
if keyResult is not None:
lexerData.add_line([f"<Keyword, {keyResult.group(0)}>"])
if sepResult:
for x in sepResult:
lexerData.add_line([f"<Separator, {x}>"])
if identResult != []:
lexerData.add_line([f"<Identifier, {identResult.group(0)}>"])
if opResult:
for x in opResult:
lexerData.add_line([f"<Operator, {x}>"])
if floatResult:
for x in floatResult:
lexerData.add_line([f"<Float, {x}>"])


Run - GUI
C:\Users\tremb\AppData\Local\Programs\Python\Python39\python.exe C:/Users/tremb/OneDrive/Documents/CS351/lexer.py
int a=15
[['<Keyword, int >']]
[['<Keyword, int >'], ['<Identifier, a>']]
[['<Keyword, int >'], ['<Identifier, a>'], ['<Operator, =>']]
[['<Keyword, int >'], ['<Identifier, a>'], ['<Operator, =>'], ['<Int, 15>']]
float b=3.14
[['<Keyword, int >'], ['<Identifier, a>'], ['<Operator, =>'], ['<Int, 15>'], ['<Keyword, float>']]
[['<Keyword, int >'], ['<Identifier, a>'], ['<Operator, =>'], ['<Int, 15>'], ['<Keyword, float>'], ['<Identifier, b>']]
[['<Keyword, int >'], ['<Identifier, a>'], ['<Operator, =>'], ['<Int, 15>'], ['<Keyword, float>'], ['<Identifier, b>'], ['<Operator, =>']]
[['<Keyword, int >'], ['<Identifier, a>'], ['<Operator, =>'], ['<Int, 15>'], ['<Keyword, float>'], ['<Identifier, b>'], ['<Operator, =>'], ['<Float, 3.14>']]
int c = a * a
[['<Keyword, int >'], ['<Identifier, a>'], ['<Operator, =>'], ['<Int, 15>'], ['<Keyword, float>'], ['<Identifier, b>'], ['<Operator, =>'], ['<Float, 3.14>'], ['<Keyword, int >']]
[['<Keyword, int >'], ['<Identifier, a>'], ['<Operator, =>'], ['<Int, 15>'], ['<Keyword, float>'], ['<Identifier, b>'], ['<Operator, =>'], ['<Float, 3.14>'], ['<Keyword, int >'], ['<Identifi
[['<Keyword, int >'], ['<Identifier, a>'], ['<Operator, =>'], ['<Int, 15>'], ['<Keyword, float>'], ['<Identifier, b>'], ['<Operator, =>'], ['<Float, 3.14>'], ['<Keyword, int >'], ['<Identifi
[['<Keyword, int >'], ['<Identifier, a>'], ['<Operator, =>'], ['<Int, 15>'], ['<Keyword, float>'], ['<Identifier, b>'], ['<Operator, =>'], ['<Float, 3.14>'], ['<Keyword, int >'], ['<Identifi
float d= 4.13
[['<Keyword, int >'], ['<Identifier, a>'], ['<Operator, =>'], ['<Int, 15>'], ['<Keyword, float>'], ['<Identifier, b>'], ['<Operator, =>'], ['<Float, 3.14>'], ['<Keyword, int >'], ['<Identifi
[['<Keyword, int >'], ['<Identifier, a>'], ['<Operator, =>'], ['<Int, 15>'], ['<Keyword, float>'], ['<Identifier, b>'], ['<Operator, =>'], ['<Float, 3.14>'], ['<Keyword, int >'], ['<Identifi
[['<Keyword, int >'], ['<Identifier, a>'], ['<Operator, =>'], ['<Int, 15>'], ['<Keyword, float>'], ['<Identifier, b>'], ['<Operator, =>'], ['<Float, 3.14>'], ['<Keyword, int >'], ['<Identifi
[['<Keyword, int >'], ['<Identifier, a>'], ['<Operator, =>'], ['<Int, 15>'], ['<Keyword, float>'], ['<Identifier, b>'], ['<Operator, =>'], ['<Float, 3.14>'], ['<Keyword, int >'], ['<Identifi
Process finished with exit code 0