

TESTCASES TEAM- 14

➤ Testcase 1: Assignment statement

```
662 def main():
663     text=" a=5+(7+8)/15 b=a+10 c= a*b + (a-b) "
664     lex = Lexer(text)
665     par = Parser(lex)

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\NITHIN CHEPURI\OneDrive\Desktop\Python for VS> & c:/Python310/python.exe "c:/Users/NITHIN CHEPURI/OneDrive/Desktop/Python for VS/interpreter.py"
ID
<__main__.Var object at 0x000001FBED046D70>
=
INTEGER
PLUS
(
INTEGER
PLUS
INTEGER
ID
<__main__.Var object at 0x000001FBED0462F0>
=
ID
MUL
ID
PLUS
(
ID
MINUS
ID
)
<__main__.Compound object at 0x000001FBED045CF0>
Run-time GLOBAL_MEMORY contents:
{'a': 6.0, 'b': 16.0, 'c': 86.0}
```

➤ Testcase 2: Arithmetic operation

```
664 def main():
665     text=" a =5 +(7+8)/15"
666     lex = Lexer(text)
667     par = Parser(lex)
668     tree = par.parse()
669     print(tree)
670     inter = Interpreter(tree)
671     result = inter.interpret()

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

ID
<__main__.Var object at 0x0000024DF2FF6D70>
=
INTEGER
PLUS
DIV
INTEGER
<__main__.Compound object at 0x0000024DF2FF6770>
Run-time GLOBAL_MEMORY contents:
{'a': 6.0}
PS C:\Users\NITHIN CHEPURI\OneDrive\Desktop\Python for VS>
```

➤ Testcase 3: Comments

```
662 def main():
663     text="--TEAM12 "
664     lex = Lexer(text)
665     par = Parser(lex)
666     tree = par.parse()
667     print(tree)
668     inter = Interpreter(tree)
669     result = inter.interpret()
670     print('Run-time GLOBAL_MEMORY contents:')
671     print(inter.GLOBAL_MEMORY)
672
673 if __name__ == '__main__':
674     main()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

Windows PowerShell
Copyright (c) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! <https://aka.ms/PSWindows>

PS C:\Users\NITHIN CHEPURI\OneDrive\Desktop\Python for VS> & C:/Python310/python.exe "c:/Users/NITHIN CHEPURI/OneDrive/Desktop/Python for VS/interpreter.py"

➤ Testcase 4: if statement

```
664 def main():
665     text=" a =5 b=0 if a>0 then b=10 end"
666     lex = Lexer(text)
667     par = Parser(lex)
668     tree = par.parse()
669     print(tree)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

```
<__main__.Var object at 0x000001BB85FB6D70>
=
INTEGER
ID
<__main__.Var object at 0x000001BB85FB6B30>
=
INTEGER
if
ID
>
INTEGER
then
ID
<__main__.Var object at 0x000001BB85FB66B0>
=
INTEGER
end
```

Run-time GLOBAL_MEMORY contents:
{'a': 5, 'b': 10}
PS C:\Users\NITHIN CHEPURI\OneDrive\Desktop\Python for VS>

➤ Testcase 5: nested if statement

```
664 def main():
665     text=" a =5 b=0 if a>0 then b=15 if b==15 then b=8 end end "
666     lex = Lexer(text)
667     par = Parser(lex)
668     tree = par.parse()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

```
ID
<__main__.Var object at 0x00000204687B6D70>
=
INTEGER
ID
<__main__.Var object at 0x00000204687B6B30>
=
INTEGER
if
ID
>
INTEGER
then
ID
<__main__.Var object at 0x00000204687B66B0>
=
INTEGER
if
ID
==
INTEGER
then
ID
<__main__.Var object at 0x00000204687B6230>
=
INTEGER
end
```

```
Run-time GLOBAL_MEMORY contents:
{'a': 5, 'b': 8}
```

➤ Testcase 6: While loop

```
663     text="a=5 b=0 while a>0 do a=a-1 b=b+1 end "
664     lex = Lexer(text)
665     par = Parser(lex)
666     tree = par.parse()
667     print(tree)
668     inter = Interpreter(tree)
669     result = inter.interpret()
670     print('Run-time GLOBAL_MEMORY contents:')
671     print(inter.GLOBAL_MEMORY)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

```
ID
<__main__.Var object at 0x00000253A5DC6D70>
=
INTEGER
ID
<__main__.Var object at 0x00000253A5DC6B30>
=
INTEGER
while
ID
>
INTEGER
do
ID
<__main__.Var object at 0x00000253A5DC66B0>
=
ID
MINUS
INTEGER
ID
<__main__.Var object at 0x00000253A5DC62F0>
=
ID
PLUS
INTEGER
end
```

```
<__main__.Compound object at 0x00000241AF845F90>  
Run-time GLOBAL MEMORY contents:  
{'a': 0, 'b': 5}
```

SYNTAX ERROR TESTCASES

➤ Testcase 1: Arithmetic operator is missing

```
664 def main():  
665     text=" a =5(6)"  
666     lex = Lexer(text)  
667     par = Parser(lex)  
668     tree = par.parse()  
669     print(tree)
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** JUPYTER

```
node = self.statement()  
File "c:\Users\NITHIN CHEPURI\OneDrive\Desktop\Python for VS\interpreter.py", line 404, in  
node=self.assignment_statement()  
File "c:\Users\NITHIN CHEPURI\OneDrive\Desktop\Python for VS\interpreter.py", line 425, in  
self.eat(ASSIGN)  
File "c:\Users\NITHIN CHEPURI\OneDrive\Desktop\Python for VS\interpreter.py", line 381, in  
self.error()  
File "c:\Users\NITHIN CHEPURI\OneDrive\Desktop\Python for VS\interpreter.py", line 373, in  
raise Exception('Invalid syntax')  
Exception: Invalid syntax  
PS C:\Users\NITHIN CHEPURI\OneDrive\Desktop\Python for VS> & C:/Python310/python.exe "c:/User
```

➤ Testcase 2

Here the error is popped because the b is not defined

```
664 def main():  
665     text=" a =b+5"  
666     lex = Lexer(text)  
667     par = Parser(lex)
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** JUPYTER

```
File "c:\Users\NITHIN CHEPURI\OneDrive\Desktop\Python for VS\TEAM_12_INTERPRETER.py", line 671, in main  
result = inter.interpret()  
File "c:\Users\NITHIN CHEPURI\OneDrive\Desktop\Python for VS\TEAM_12_INTERPRETER.py", line 660, in interpret  
return self.visit(tree)  
File "c:\Users\NITHIN CHEPURI\OneDrive\Desktop\Python for VS\TEAM_12_INTERPRETER.py", line 576, in visit  
return visitor(node)  
File "c:\Users\NITHIN CHEPURI\OneDrive\Desktop\Python for VS\TEAM_12_INTERPRETER.py", line 625, in visit_Compound  
self.visit(child)  
File "c:\Users\NITHIN CHEPURI\OneDrive\Desktop\Python for VS\TEAM_12_INTERPRETER.py", line 576, in visit  
return visitor(node)  
File "c:\Users\NITHIN CHEPURI\OneDrive\Desktop\Python for VS\TEAM_12_INTERPRETER.py", line 629, in visit_Assign  
var value = self.visit(node.right)  
File "c:\Users\NITHIN CHEPURI\OneDrive\Desktop\Python for VS\TEAM_12_INTERPRETER.py", line 576, in visit  
return visitor(node)  
File "c:\Users\NITHIN CHEPURI\OneDrive\Desktop\Python for VS\TEAM_12_INTERPRETER.py", line 591, in visit_BinOp  
return self.visit(node.left) + self.visit(node.right)  
File "c:\Users\NITHIN CHEPURI\OneDrive\Desktop\Python for VS\TEAM_12_INTERPRETER.py", line 576, in visit  
return visitor(node)  
File "c:\Users\NITHIN CHEPURI\OneDrive\Desktop\Python for VS\TEAM_12_INTERPRETER.py", line 636, in visit_Var  
raise NameError(repr(var_name))  
NameError: 'b'
```


➤ Testcase 3

Here “then” token is missing that’s why it showed “Invalid syntax”.

```
664 def main():
665     text=" a =5 b=0 if a>0 b=15 "
666     lex = Lexer(text)
667     par = Parser(lex)
668     tree = par.parse()
669     print(tree)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

```
ID
>
INTEGER
ID
Traceback (most recent call last):
  File "c:\Users\NITHIN CHEPURI\OneDrive\Desktop\Python for VS\TEAM_12_INTERPRETER.py", line 676, in <module>
    main()
  File "c:\Users\NITHIN CHEPURI\OneDrive\Desktop\Python for VS\TEAM_12_INTERPRETER.py", line 668, in main
    self.error()
  File "c:\Users\NITHIN CHEPURI\OneDrive\Desktop\Python for VS\TEAM_12_INTERPRETER.py", line 371, in error
    raise Exception('Invalid syntax')
Exception: Invalid syntax
PS C:\Users\NITHIN CHEPURI\OneDrive\Desktop\Python for VS> █
```

➤ Testcase 4

Here “do” is missing that’s why it showed “Invalid syntax”

```
664 def main():
665     text=" a = 5 b=0 while a>0 b=b+1"
666     lex = Lexer(text)
667     par = Parser(lex)
668     tree = par.parse()
669     print(tree)
670     inter = Interpreter(tree)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

```
nodes = self.statement_list()
File "c:\Users\NITHIN CHEPURI\OneDrive\Desktop\Python for VS\TEAM_12_INTERPRETER.py", line 396, in statement_list
    results.append(self.statement())
File "c:\Users\NITHIN CHEPURI\OneDrive\Desktop\Python for VS\TEAM_12_INTERPRETER.py", line 406, in statement
    node=self.while_statement()
File "c:\Users\NITHIN CHEPURI\OneDrive\Desktop\Python for VS\TEAM_12_INTERPRETER.py", line 447, in while_statement
    self.eat(DO)
File "c:\Users\NITHIN CHEPURI\OneDrive\Desktop\Python for VS\TEAM_12_INTERPRETER.py", line 379, in eat
    self.error()
File "c:\Users\NITHIN CHEPURI\OneDrive\Desktop\Python for VS\TEAM_12_INTERPRETER.py", line 371, in error
    raise Exception('Invalid syntax')
Exception: Invalid syntax
```