

Project Three : Semantic Analysis Output

Jordan Murray

Jordan.Murray1@Marist.edu

May 4, 2021

NOTE: This document excludes the lexer and parser output for conciseness

1 TEST CASE 1

1.1 PROGRAM

```
{
int a
boolean b
{
string c
a = 5
b = true /* no comment */
c = "inta"
print(c)
}
print(b)
print(a)
}$
```

1.2 OUTPUT

SEMANTIC ANALYZER: Analyzing program 1

SEMANTIC ANALYZER: BEGIN SCOPE - 0

SEMANTIC ANALYZER: DECLARATION - Variable 'a' declared and added to Symbol table

SEMANTIC ANALYZER: DECLARATION - Variable 'b' declared and added to Symbol table

SEMANTIC ANALYZER: BEGIN SCOPE - 1

SEMANTIC ANALYZER: DECLARATION - Variable 'c' declared and added to Symbol table

SEMANTIC ANALYZER: INITIALIZATION - Variable 'a' initialized

SEMANTIC ANALYZER: TYPE CHECK - Variable 'a' verified

SEMANTIC ANALYZER: INITIALIZATION - Variable 'b' initialized
 SEMANTIC ANALYZER: TYPE CHECK - Variable 'b' verified
 SEMANTIC ANALYZER: INITIALIZATION - Variable 'c' initialized
 SEMANTIC ANALYZER: TYPE CHECK - Variable 'c' verified
 SEMANTIC ANALYZER: CLOSE SCOPE - 1
 SEMANTIC ANALYZER: CLOSE SCOPE - 0
 WARNING: Variable 'a' in scope 0 is declared and used but not initialized
 WARNING: Variable 'b' in scope 0 is declared and used but not initialized
 SEMANTIC ANALYZER: Program 1 finished with 0 error(s) and 2 warning(s)

AST for program 1

```

<Block>
- <Variable Declaration>
- [int]
- [a]
- <Variable Declaration>
- [boolean]
- [b]
- <Block>
- <Variable Declaration>
- [string]
- [c]
- <Assignment Statement>
- [a]
- [5]
- <Assignment Statement>
- [b]
- [true]
- <Assignment Statement>
- [c]
- [inta]
- <Print Statement>
- [c]
- <Print Statement>
- [b]
- <Print Statement>
- [a]
  
```

Program 1 Symbol Table

Name	Type	Scope	Line
a	int	0	2
b	boolean	0	3
c	string	1	5

1.3 COMMENTS

This is a test program that passes, but has 2 warnings due to the initialization variables a and b in a different scopes.

2 TEST CASE 2

2.1 PROGRAM

```
{  
int a  
{  
boolean b  
a = 1  
}  
print(b)  
}$
```

2.2 OUTPUT

```
SEMANTIC ANALYZER: Analyzing program 2  
SEMANTIC ANALYZER: BEGIN SCOPE - 0  
SEMANTIC ANALYZER: DECLARATION - Variable 'a' declared and added to Symbol table  
SEMANTIC ANALYZER: BEGIN SCOPE - 1  
SEMANTIC ANALYZER: DECLARATION - Variable 'b' declared and added to Symbol table  
SEMANTIC ANALYZER: INITIALIZATION - Variable 'a' initialized  
SEMANTIC ANALYZER: TYPE CHECK - Variable 'a' verified  
SEMANTIC ANALYZER: CLOSE SCOPE - 1  
ERROR: Variable 'b' on line 21 is not declared in scope  
WARNING: Variable 'a' in scope 0 is declared but not initialized or used  
WARNING: Variable 'b' in scope 1 is declared but not initialized or used  
SEMANTIC ANALYZER: Program 2 finished with 1 error(s) and 2 warning(s)
```

SEMANTIC ANALYZER: AST and Symbol table skipped due to Semantic Analysis Error(s)

2.3 COMMENTS

This program tests the declaration of a var in a scope that is inaccessible. 1 error is thrown and 2 warnings are thrown for the vars that are declared but not initialized or used

3 TEST CASE 3

3.1 PROGRAM

```
{  
int a  
a = true  
}$
```

3.2 OUTPUT

SEMANTIC ANALYZER: Analyzing program 3
SEMANTIC ANALYZER: BEGIN SCOPE - 0
SEMANTIC ANALYZER: DECLARATION - Variable 'a' declared and added to Symbol table
SEMANTIC ANALYZER: INITIALIZATION - Variable 'a' initialized
ERROR: Variable 'a' on line 26 assigned wrong type
WARNING: Variable 'a' in scope 0 is declared and initialized but not used
SEMANTIC ANALYZER: Program 3 finished with 1 error(s) and 1 warning(s)

SEMANTIC ANALYZER: AST and Symbol table skipped due to Semantic Analysis Error(s)

3.3 COMMENTS

This program tests assigning a var of type int a value of type boolean, and throws an error

4 TEST CASE 4

4.1 PROGRAM

```
{  
int a  
string a  
}$
```

4.2 OUTPUT

SEMANTIC ANALYZER: Analyzing program 4
SEMANTIC ANALYZER: BEGIN SCOPE - 0
SEMANTIC ANALYZER: DECLARATION - Variable 'a' declared and added to Symbol table
ERROR: Variable 'a' on line 31 has already been declared in scope
WARNING: Variable 'a' in scope 0 is declared but not initialized or used
SEMANTIC ANALYZER: Program 4 finished with 1 error(s) and 1 warning(s)

SEMANTIC ANALYZER: AST and Symbol table skipped due to Semantic Analysis Error(s)

4.3 COMMENTS

This program tests when the same id is re-declared in scope, even with a different type

5 TEST CASE 5

5.1 PROGRAM

```
{int a}$
```

5.2 OUTPUT

```
SEMANTIC ANALYZER: Analyzing program 5
SEMANTIC ANALYZER: BEGIN SCOPE - 0
SEMANTIC ANALYZER: DECLARATION - Variable 'a' declared and added to Symbol table
SEMANTIC ANALYZER: CLOSE SCOPE - 0
WARNING: Variable 'a' in scope 0 is declared but not initialized or used
SEMANTIC ANALYZER: Program 5 finished with 0 error(s) and 1 warning(s)
```

AST for program 5

```
<Block>
-<Variable Declaration>
-[int]
-[a]
```

Program 5 Symbol Table

Name	Type	Scope	Line
a	int	0	34

5.3 COMMENTS

This program tests basic int variable declaration and throws a warning because it is not initialized or used

6 TEST CASE 6

6.1 PROGRAM

```
{
int a
{
string a
}
}$
```

6.2 OUTPUT

```
SEMANTIC ANALYZER: Analyzing program 6
SEMANTIC ANALYZER: BEGIN SCOPE - 0
SEMANTIC ANALYZER: DECLARATION - Variable 'a' declared and added to Symbol table
SEMANTIC ANALYZER: BEGIN SCOPE - 1
SEMANTIC ANALYZER: DECLARATION - Variable 'a' declared and added to Symbol table
SEMANTIC ANALYZER: CLOSE SCOPE - 1
SEMANTIC ANALYZER: CLOSE SCOPE - 0
WARNING: Variable 'a' in scope 0 is declared but not initialized or used
WARNING: Variable 'a' in scope 1 is declared but not initialized or used
SEMANTIC ANALYZER: Program 6 finished with 0 error(s) and 2 warning(s)
```

AST for program 6

```
<Block>
-<Variable Declaration>
-[int]
-[a]
-<Block>
—<Variable Declaration>
—[string]
—[a]
```

Program 6 Symbol Table

Name	Type	Scope	Line
a	int	0	37
a	string	1	39

6.3 COMMENTS

This program tests variable declaration using the same id, but in different scopes. It throws warnings for no initialization/ use for both declarations.

7 TEST CASE 7

7.1 PROGRAM

```
{
boolean a
{
int a
a = 5
}
}$
```

7.2 OUTPUT

```
SEMANTIC ANALYZER: Analyzing program 7
SEMANTIC ANALYZER: BEGIN SCOPE - 0
SEMANTIC ANALYZER: DECLARATION - Variable 'a' declared and added to Symbol table
SEMANTIC ANALYZER: BEGIN SCOPE - 1
SEMANTIC ANALYZER: DECLARATION - Variable 'a' declared and added to Symbol table
SEMANTIC ANALYZER: INITIALIZATION - Variable 'a' initialized
SEMANTIC ANALYZER: TYPE CHECK - Variable 'a' verified
SEMANTIC ANALYZER: CLOSE SCOPE - 1
SEMANTIC ANALYZER: CLOSE SCOPE - 0
WARNING: Variable 'a' in scope 0 is declared but not initialized or used
WARNING: Variable 'a' in scope 1 is declared and initialized but not used
SEMANTIC ANALYZER: Program 7 finished with 0 error(s) and 2 warning(s)
```

AST for program 7

```
<Block>
-<Variable Declaration>
-[boolean]
-[a]
-<Block>
--<Variable Declaration>
--[int]
--[a]
--<Assignment Statement>
--[a]
--[5]
```

Program 7 Symbol Table

Name	Type	Scope	Line
a	boolean	0	44
a	int	1	46

7.3 COMMENTS

This program is similar to program 6, but adds a declaration to test that they get different warning messages

8 TEST CASE 8

8.1 PROGRAM

```
{
{
int a
a = 5
}
print(a)
}$
```

8.2 OUTPUT

```
SEMANTIC ANALYZER: Analyzing program 8
SEMANTIC ANALYZER: BEGIN SCOPE - 0
SEMANTIC ANALYZER: BEGIN SCOPE - 1
SEMANTIC ANALYZER: DECLARATION - Variable 'a' declared and added to Symbol table
SEMANTIC ANALYZER: INITIALIZATION - Variable 'a' initialized
SEMANTIC ANALYZER: TYPE CHECK - Variable 'a' verified
SEMANTIC ANALYZER: CLOSE SCOPE - 1
ERROR: Variable 'a' on line 56 is not declared in scope
WARNING: Variable 'a' in scope 1 is declared and initialized but not used
SEMANTIC ANALYZER: Program 8 finished with 1 error(s) and 1 warning(s)
```

SEMANTIC ANALYZER: AST and Symbol table skipped due to Semantic Analysis Error(s)

8.3 COMMENTS

This program tests the use of a variable that is declared and initialized in one scope, but used in a different scope.

9 TEST CASE 9

9.1 PROGRAM

```
{
int a
a = 3
{
a = 5
}
print(a)
}$
```

9.2 OUTPUT

```
SEMANTIC ANALYZER: Analyzing program 9
SEMANTIC ANALYZER: BEGIN SCOPE - 0
SEMANTIC ANALYZER: DECLARATION - Variable 'a' declared and added to Symbol table
SEMANTIC ANALYZER: INITIALIZATION - Variable 'a' initialized
SEMANTIC ANALYZER: TYPE CHECK - Variable 'a' verified
SEMANTIC ANALYZER: BEGIN SCOPE - 1
SEMANTIC ANALYZER: INITIALIZATION - Variable 'a' initialized
SEMANTIC ANALYZER: TYPE CHECK - Variable 'a' verified
SEMANTIC ANALYZER: CLOSE SCOPE - 1
SEMANTIC ANALYZER: CLOSE SCOPE - 0
SEMANTIC ANALYZER: Program 9 finished with 0 error(s) and 0 warning(s)
```

AST for program 9

```
<Block>
-<Variable Declaration>
-[int]
-[a]
-<Assignment Statement>
-[a]
-[3]
-<Block>
--<Assignment Statement>
-[a]
-[5]
-<Print Statement>
```


—[a]

Program 9 Symbol Table

Name	Type	Scope	Line
a	int	0	60

9.3 COMMENTS

This program tests a re-initialization of a var declared and initialized in a parent scope.

10 TEST CASE 10

10.1 PROGRAM

```
{int a {int b}{print(b)}}$
```

10.2 OUTPUT

```
SEMANTIC ANALYZER: Analyzing program 10
SEMANTIC ANALYZER: BEGIN SCOPE - 0
SEMANTIC ANALYZER: DECLARATION - Variable 'a' declared and added to Symbol table
SEMANTIC ANALYZER: BEGIN SCOPE - 1
SEMANTIC ANALYZER: DECLARATION - Variable 'b' declared and added to Symbol table
SEMANTIC ANALYZER: CLOSE SCOPE - 1
SEMANTIC ANALYZER: BEGIN SCOPE - 2
ERROR: Variable 'b' on line 68 is not declared in scope
WARNING: Variable 'a' in scope 0 is declared but not initialized or used
WARNING: Variable 'b' in scope 1 is declared but not initialized or used
SEMANTIC ANALYZER: Program 10 finished with 1 error(s) and 2 warning(s)
```

SEMANTIC ANALYZER: AST and Symbol table skipped due to Semantic Analysis Error(s)

10.3 COMMENTS

This test case tests jumping back multiple scopes. Since there is no scope tree, it tests the use of the numScopes counter var to work properly. This should fail since b is declared in a different scope than it is used.

11 TEST CASE 11

11.1 PROGRAM

```
{
int a
a = a
string b
```

```
a = b
}$
```

11.2 OUTPUT

```
SEMANTIC ANALYZER: Analyzing program 11
SEMANTIC ANALYZER: BEGIN SCOPE - 0
SEMANTIC ANALYZER: DECLARATION - Variable 'a' declared and added to Symbol table
SEMANTIC ANALYZER: INITIALIZATION - Variable 'a' initialized
SEMANTIC ANALYZER: TYPE CHECK - Variable 'a' verified
SEMANTIC ANALYZER: DECLARATION - Variable 'b' declared and added to Symbol table
SEMANTIC ANALYZER: INITIALIZATION - Variable 'a' initialized
ERROR: Variable 'a' on line 74 is assigned wrong type
WARNING: Variable 'b' in scope 0 is declared and used but not initialized
SEMANTIC ANALYZER: Program 11 finished with 1 error(s) and 1 warning(s)
```

SEMANTIC ANALYZER: AST and Symbol table skipped due to Semantic Analysis Error(s)

11.3 COMMENTS

This program tests the assignment of an id using another id of a different type, which throws an error and a warning because the var is not initialized

12 TEST CASE 12

12.1 PROGRAM

```
{boolean a while ("string" == (a == (a != "int"))) {}}$
```

12.2 OUTPUT

```
SEMANTIC ANALYZER: Analyzing program 12
SEMANTIC ANALYZER: BEGIN SCOPE - 0
SEMANTIC ANALYZER: DECLARATION - Variable 'a' declared and added to Symbol table
ERROR: Variable 'a' on line 77 cannot be compared to type: string
WARNING: Variable 'a' in scope 0 is declared and used but not initialized
SEMANTIC ANALYZER: Program 12 finished with 1 error(s) and 1 warning(s)
```

SEMANTIC ANALYZER: AST and Symbol table skipped due to Semantic Analysis Error(s)

12.3 COMMENTS

This test case tests to see how nested boolean expressions would be handled. An error is thrown because a boolean is being compared to a string

13 TEST CASE 13

13.1 PROGRAM

```
{
int a
a = 1
print(a)
boolean b
b = true
print(b)
{
int a
a = 2
print(a)
}
{
int a
a = 3
print(a)
}
string s
s = "stra"
print(s)
s = "strb"
print(s)
if (a != 5) {
print("true")
}
if (a == 5) {
print("false")
}
}$
```

13.2 OUTPUT

```
SEMANTIC ANALYZER: Analyzing program 13
SEMANTIC ANALYZER: BEGIN SCOPE - 0
SEMANTIC ANALYZER: DECLARATION - Variable 'a' declared and added to Symbol table
SEMANTIC ANALYZER: INITIALIZATION - Variable 'a' initialized
SEMANTIC ANALYZER: TYPE CHECK - Variable 'a' verified
SEMANTIC ANALYZER: DECLARATION - Variable 'b' declared and added to Symbol table
SEMANTIC ANALYZER: INITIALIZATION - Variable 'b' initialized
SEMANTIC ANALYZER: TYPE CHECK - Variable 'b' verified
SEMANTIC ANALYZER: BEGIN SCOPE - 1
SEMANTIC ANALYZER: DECLARATION - Variable 'a' declared and added to Symbol table
SEMANTIC ANALYZER: INITIALIZATION - Variable 'a' initialized
SEMANTIC ANALYZER: TYPE CHECK - Variable 'a' verified
SEMANTIC ANALYZER: CLOSE SCOPE - 1
SEMANTIC ANALYZER: BEGIN SCOPE - 2
SEMANTIC ANALYZER: DECLARATION - Variable 'a' declared and added to Symbol table
```

SEMANTIC ANALYZER: INITIALIZATION - Variable 'a' initialized
 SEMANTIC ANALYZER: TYPE CHECK - Variable 'a' verified
 SEMANTIC ANALYZER: CLOSE SCOPE - 2
 SEMANTIC ANALYZER: DECLARATION - Variable 's' declared and added to Symbol table
 SEMANTIC ANALYZER: INITIALIZATION - Variable 's' initialized
 SEMANTIC ANALYZER: TYPE CHECK - Variable 's' verified
 SEMANTIC ANALYZER: INITIALIZATION - Variable 's' initialized
 SEMANTIC ANALYZER: TYPE CHECK - Variable 's' verified
 SEMANTIC ANALYZER: TYPE CHECK - Variable 'a' verified
 SEMANTIC ANALYZER: BEGIN SCOPE - 3
 SEMANTIC ANALYZER: CLOSE SCOPE - 3
 SEMANTIC ANALYZER: TYPE CHECK - Variable 'a' verified
 SEMANTIC ANALYZER: BEGIN SCOPE - 4
 SEMANTIC ANALYZER: CLOSE SCOPE - 4
 SEMANTIC ANALYZER: CLOSE SCOPE - 0
 SEMANTIC ANALYZER: Program 13 finished with 0 error(s) and 0 warning(s)

AST for program 13

```

<Block>
- <Variable Declaration>
- [int]
- [a]
- <Assignment Statement>
- [a]
- [1]
- <Print Statement>
- [a]
- <Variable Declaration>
- [boolean]
- [b]
- <Assignment Statement>
- [b]
- [true]
- <Print Statement>
- [b]
- <Block>
- <Variable Declaration>
- [int]
- [a]
- <Assignment Statement>
- [a]
- [2]
- <Print Statement>
- [a]
- <Block>
- <Variable Declaration>
- [int]
- [a]
- <Assignment Statement>
- [a]
- [3]
  
```

```

——<Print Statement>
——[a]
——<Variable Declaration>
——[string]
——[s]
——<Assignment Statement>
——[s]
——[stra]
——<Print Statement>
——[s]
——<Assignment Statement>
——[s]
——[strb]
——<Print Statement>
——[s]
——<If Statement>
——[a]
——[!=]
——[5]
——<Block>
——<Print Statement>
——[true]
——<If Statement>
——[a]
——[==]
——[5]
——<Block>
——<Print Statement>
——[false]

```

Program 13 Symbol Table

Name	Type	Scope	Line
a	int	0	80
b	boolean	0	83
a	int	1	87
a	int	2	92
s	string	0	96

13.3 COMMENTS

This is a long test case (taken from one of the Compilers in the hall of fame) that shows a successful semantic analysis on a more complex program. It tests multiple scopes and the declaration of variables with the same id in multiple different scopes.