

Semantic Analysis Testing Results

Emily Doran

Emily.Doran1@Marist.edu

May 2, 2021

BASIC TEST CASES (VERBOSE MODE)

Input text:

```
{}$
{{{({})}}}}}$
{{{({})}} /* comments are ignored */ }}}}
{/* comments are still ignored */ int @}$
{
    int a
    boolean f
    a = a
    string b
    a = b
}$
{int int}$

{int a
a = 3 + 2}$
```

Output:

```
INFO  Lexer - Lexing program 1...
DEBUG Lexer - T_L_BRACE [ { ] found at (1:1)
DEBUG Lexer - T_R_BRACE [ } ] found at (1:2)
DEBUG Lexer - T_EOP [ $ ] found at (1:3)
INFO  Lexer - Lex completed with 0 errors

PARSER: Parsing program 1 ...
PARSER: parse()
PARSER: parseProgram()
PARSER: parseBlock()
PARSER: parseStatementList()
PARSER: Parse completed successfully
```

CST for program 1 ...

```
<Program>
-<Block>
--[{}
--[StatementList]
--[]]
-[$]
```

SEMANTIC ANALYSIS: Beginning Semantic Analysis on Program 1 ...

SEMANTIC ANALYSIS: New Scope [0] has been entered at line: 1.

Program 1 Semantic Analysis produced 0 error(s) and 0 warning(s).

AST for program 1 ...

[BLOCK]

Program 1 Symbol Table

```
-----
Name  Type      Scope  Line
-----
```

INFO Lexer - Lexing program 2...

```
DEBUG Lexer - T_L_BRACE [ { ] found at (2:1)
DEBUG Lexer - T_L_BRACE [ { ] found at (2:2)
DEBUG Lexer - T_L_BRACE [ { ] found at (2:3)
DEBUG Lexer - T_L_BRACE [ { ] found at (2:4)
DEBUG Lexer - T_L_BRACE [ { ] found at (2:5)
DEBUG Lexer - T_L_BRACE [ { ] found at (2:6)
DEBUG Lexer - T_R_BRACE [ } ] found at (2:7)
DEBUG Lexer - T_R_BRACE [ } ] found at (2:8)
DEBUG Lexer - T_R_BRACE [ } ] found at (2:9)
DEBUG Lexer - T_R_BRACE [ } ] found at (2:10)
DEBUG Lexer - T_R_BRACE [ } ] found at (2:11)
DEBUG Lexer - T_R_BRACE [ } ] found at (2:12)
DEBUG Lexer - T_EOP [ $ ] found at (2:13)
INFO  Lexer - Lex completed with 0 errors
```

PARSER: Parsing program 2 ...

```
PARSER: parse()
PARSER: parseProgram()
PARSER: parseBlock()
PARSER: parseStatementList()
PARSER: parseStatement()
PARSER: parseBlock()
PARSER: parseStatementList()
PARSER: parseStatement()
PARSER: parseBlock()
PARSER: parseStatementList()
PARSER: parseStatement()
PARSER: parseBlock()
PARSER: parseStatementList()
PARSER: parseStatement()
```

```

PARSER: parseBlock()
PARSER: parseStatementList()
PARSER: parseStatementList()
PARSER: parseStatementList()
PARSER: parseStatementList()
PARSER: parseStatementList()
PARSER: parseStatementList()
PARSER: Parse completed successfully

```

CST for program 2 ...

```

<Program>
-<Block>
--[{}
--<StatementList>
---<Statement>
----<Block>
-----[{}
-----<StatementList>
-----<Statement>
-----<Block>
-----[{}
-----<StatementList>
-----<Statement>
-----<Block>
-----[{}
-----<StatementList>
-----<Statement>
-----<Block>
-----[{}
-----<StatementList>
-----<Statement>
-----<Block>
-----[{}
-----[StatementList]
-----[}]
-----[}]
-----[}]
-----[}]
-----[}]
--[}]
-[{}]
-[$]

```

```

SEMANTIC ANALYSIS: Beginning Semantic Analysis on Program 2 ...
SEMANTIC ANALYSIS: New Scope [ 0 ] has been entered at line: 2.
SEMANTIC ANALYSIS: New Scope [ 1 ] has been entered at line: 2.
SEMANTIC ANALYSIS: Scope [ 1 ] parent scope has been set to [ 0 ] at line: 2.
SEMANTIC ANALYSIS: New Scope [ 2 ] has been entered at line: 2.
SEMANTIC ANALYSIS: Scope [ 2 ] parent scope has been set to [ 1 ] at line: 2.
SEMANTIC ANALYSIS: New Scope [ 3 ] has been entered at line: 2.
SEMANTIC ANALYSIS: Scope [ 3 ] parent scope has been set to [ 2 ] at line: 2.
SEMANTIC ANALYSIS: New Scope [ 4 ] has been entered at line: 2.
SEMANTIC ANALYSIS: Scope [ 4 ] parent scope has been set to [ 3 ] at line: 2.
SEMANTIC ANALYSIS: New Scope [ 5 ] has been entered at line: 2.
SEMANTIC ANALYSIS: Scope [ 5 ] parent scope has been set to [ 4 ] at line: 2.
SEMANTIC ANALYSIS: Exiting scope [ 5 ] and entering scope [ 4 ] at line: 2.
SEMANTIC ANALYSIS: Exiting scope [ 4 ] and entering scope [ 3 ] at line: 2.

```

```

SEMANTIC ANALYSIS: Exiting scope [ 3 ] and entering scope [ 2 ] at line: 2.
SEMANTIC ANALYSIS: Exiting scope [ 2 ] and entering scope [ 1 ] at line: 2.
SEMANTIC ANALYSIS: Exiting scope [ 1 ] and entering scope [ 0 ] at line: 2.

```

Program 2 Semantic Analysis produced 0 error(s) and 0 warning(s).

AST for program 2 ...

```

<BLOCK>
-<BLOCK>
--<BLOCK>
---<BLOCK>
----<BLOCK>
-----[BLOCK]

```

Program 2 Symbol Table

```

-----
Name  Type      Scope  Line
-----

```

```

INFO  Lexer - Lexing program 3...
DEBUG Lexer - T_L_BRACE [ { ] found at (3:1)
DEBUG Lexer - T_L_BRACE [ { ] found at (3:2)
DEBUG Lexer - T_L_BRACE [ { ] found at (3:3)
DEBUG Lexer - T_L_BRACE [ { ] found at (3:4)
DEBUG Lexer - T_L_BRACE [ { ] found at (3:5)
DEBUG Lexer - T_L_BRACE [ { ] found at (3:6)
DEBUG Lexer - T_R_BRACE [ } ] found at (3:7)
DEBUG Lexer - T_R_BRACE [ } ] found at (3:8)
DEBUG Lexer - T_R_BRACE [ } ] found at (3:9)
DEBUG Lexer - T_R_BRACE [ } ] found at (3:38)
DEBUG Lexer - T_R_BRACE [ } ] found at (3:39)
DEBUG Lexer - T_R_BRACE [ } ] found at (3:40)
DEBUG Lexer - T_R_BRACE [ } ] found at (3:41)
DEBUG Lexer - T_EOP [ $ ] found at (3:42)
INFO  Lexer - Lex completed with 0 errors

```

PARSER: Parsing program 3 ...

```

PARSER: parse()
PARSER: parseProgram()
PARSER: parseBlock()
PARSER: parseStatementList()
PARSER: parseStatement()
PARSER: parseBlock()
PARSER: parseStatementList()
PARSER: parseStatement()
PARSER: parseBlock()
PARSER: parseStatementList()
PARSER: parseStatement()
PARSER: parseBlock()
PARSER: parseStatementList()
PARSER: parseStatement()
PARSER: parseBlock()

```

```
PARSER: parseStatementList()
PARSER: parseStatementList()
PARSER: parseStatementList()
PARSER: parseStatementList()
PARSER: parseStatementList()
PARSER: parseStatementList()
PARSER: ERROR: Expected [EOP] got '}' on line 3
PARSER: Parse failed with 1 error(s)
```

CST for program 3: Skipped due to PARSER error(s)

Semantic Analysis for program 3: Skipped due to PARSER error(s)

AST for program 3: Skipped due to PARSER error(s)

Symbol Table for program 3: Skipped due to PARSER error(s)

```
INFO  Lexer - Lexing program 4...
DEBUG Lexer - T_L_BRACE [ { ] found at (4:1)
DEBUG Lexer - T_VARIABLE_TYPE [ int ] found at (4:36)
ERROR Lexer - Error: 4:40 Unrecognized Token: @
DEBUG Lexer - T_R_BRACE [ } ] found at (4:41)
DEBUG Lexer - T_EOP [ $ ] found at (4:42)
ERROR Lexer - Lex failed with 1 error(s)
```

PARSER: Skipped due to LEXER error(s)

CST for program 4: Skipped due to LEXER error(s)

Semantic Analysis for program 4: Skipped due to LEXER error(s)

AST for program 4: Skipped due to LEXER error(s)

Symbol Table for program 4: Skipped due to LEXER error(s)

```
INFO  Lexer - Lexing program 5...
DEBUG Lexer - T_L_BRACE [ { ] found at (5:1)
DEBUG Lexer - T_VARIABLE_TYPE [ int ] found at (6:3)
DEBUG Lexer - T_ID [ a ] found at (6:7)
DEBUG Lexer - T_VARIABLE_TYPE [ boolean ] found at (7:3)
DEBUG Lexer - T_ID [ f ] found at (7:11)
DEBUG Lexer - T_ID [ a ] found at (8:3)
DEBUG Lexer - T_ASSIGN_OP [ = ] found at (8:5)
DEBUG Lexer - T_ID [ a ] found at (8:7)
DEBUG Lexer - T_VARIABLE_TYPE [ string ] found at (9:3)
DEBUG Lexer - T_ID [ b ] found at (9:10)
DEBUG Lexer - T_ID [ a ] found at (10:3)
DEBUG Lexer - T_ASSIGN_OP [ = ] found at (10:5)
DEBUG Lexer - T_ID [ b ] found at (10:7)
DEBUG Lexer - T_R_BRACE [ } ] found at (11:1)
DEBUG Lexer - T_EOP [ $ ] found at (11:2)
INFO  Lexer - Lex completed with 0 errors
```

PARSER: Parsing program 5 ...

```

PARSER: parse()
PARSER: parseProgram()
PARSER: parseBlock()
PARSER: parseStatementList()
PARSER: parseStatement()
PARSER: parseVarDecl()
PARSER: parseType()
PARSER: parseStatementList()
PARSER: parseStatement()
PARSER: parseVarDecl()
PARSER: parseType()
PARSER: parseStatementList()
PARSER: parseStatement()
PARSER: parseAssignStatement()
PARSER: parseExpr()
PARSER: parseStatementList()
PARSER: parseStatement()
PARSER: parseVarDecl()
PARSER: parseType()
PARSER: parseStatementList()
PARSER: parseStatement()
PARSER: parseAssignStatement()
PARSER: parseExpr()
PARSER: parseStatementList()
PARSER: Parse completed successfully

```

CST for program 5 ...

```

<Program>
-<Block>
--[{}
--<StatementList>
---<Statement>
----<VarDecl>
-----<Type>
-----[int]
-----<Id>
-----[a]
---<StatementList>
----<Statement>
-----<VarDecl>
-----<Type>
-----[boolean]
-----<Id>
-----[f]
----<StatementList>
-----<Statement>
-----<AssignStatement>
-----<Id>
-----[a]
-----[=]
-----<Expression>
-----<Id>
-----[a]
-----<StatementList>
-----<Statement>
-----<VarDecl>

```

```

-----<Type>
-----[string]
-----<Id>
-----[b]
-----<StatementList>
-----<Statement>
-----<AssignStatement>
-----<Id>
-----[a]
-----[=]
-----<Expression>
-----<Id>
-----[b]
--[]]
-[$]

```

```

SEMANTIC ANALYSIS: Beginning Semantic Analysis on Program 5 ...
SEMANTIC ANALYSIS: New Scope [ 0 ] has been entered at line: 5.
SEMANTIC ANALYSIS: Variable [ a ] has been declared at (6:3)
SEMANTIC ANALYSIS: Variable [ f ] has been declared at (7:3)
SEMANTIC ANALYSIS: Variable [ a ] has been used at (8:7)
SEMANTIC ANALYSIS: Variable [ b ] has been declared at (9:3)
SEMANTIC ANALYSIS: Variable [ b ] has been used at (10:7)
SEMANTIC ANALYSIS: ERROR: Incorrect Type Comparison - Variable [ a ] of type [ int
] was compared to type [ string ] at (10:7).
SEMANTIC ANALYSIS: WARNING: Variable [ b ] is declared and used but never
initialized.
SEMANTIC ANALYSIS: WARNING: Variable [ a ] is declared and used but never
initialized.
SEMANTIC ANALYSIS: WARNING: Variable [ f ] is declared but never initialized or
used.

```

Program 5 Semantic Analysis produced 1 error(s) and 3 warning(s).

AST for program 5: Skipped due to SEMANTIC ANALYSIS error(s)

Symbol Table for program 5: Skipped due to SEMANTIC ANALYSIS error(s)

```

INFO  Lexer - Lexing program 6...
DEBUG Lexer - T_L_BRACE [ { ] found at (12:1)
DEBUG Lexer - T_VARIABLE_TYPE [ int ] found at (12:2)
DEBUG Lexer - T_VARIABLE_TYPE [ int ] found at (12:6)
DEBUG Lexer - T_R_BRACE [ } ] found at (12:9)
DEBUG Lexer - T_EOP [ $ ] found at (12:10)
INFO  Lexer - Lex completed with 0 errors

```

```

PARSER: Parsing program 6 ...
PARSER: parse()
PARSER: parseProgram()
PARSER: parseBlock()
PARSER: parseStatementList()
PARSER: parseStatement()
PARSER: parseVarDecl()
PARSER: parseType()
PARSER: ERROR: Expected [id] got 'int' on line 12

```

```

PARSER: Parse failed with 1 error(s)

CST for program 6: Skipped due to PARSER error(s)

Semantic Analysis for program 6: Skipped due to PARSER error(s)

AST for program 6: Skipped due to PARSER error(s)

Symbol Table for program 6: Skipped due to PARSER error(s)

```

```

INFO  Lexer - Lexing program 7...
DEBUG Lexer - T_L_BRACE [ { ] found at (14:1)
DEBUG Lexer - T_VARIABLE_TYPE [ int ] found at (14:2)
DEBUG Lexer - T_ID [ a ] found at (14:6)
DEBUG Lexer - T_ID [ a ] found at (15:1)
DEBUG Lexer - T_ASSIGN_OP [ = ] found at (15:3)
DEBUG Lexer - T_DIGIT [ 3 ] found at (15:5)
DEBUG Lexer - T_ADDITION_OP [ + ] found at (15:7)
DEBUG Lexer - T_DIGIT [ 2 ] found at (15:9)
DEBUG Lexer - T_R_BRACE [ } ] found at (15:10)
DEBUG Lexer - T_EOP [ $ ] found at (15:11)
INFO  Lexer - Lex completed with 0 errors

```

```

PARSER: Parsing program 7 ...
PARSER: parse()
PARSER: parseProgram()
PARSER: parseBlock()
PARSER: parseStatementList()
PARSER: parseStatement()
PARSER: parseVarDecl()
PARSER: parseType()
PARSER: parseStatementList()
PARSER: parseStatement()
PARSER: parseAssignStatement()
PARSER: parseExpr()
PARSER: parseIntExpr()
PARSER: parseExpr()
PARSER: parseIntExpr()
PARSER: parseStatementList()
PARSER: Parse completed successfully

```

```

CST for program 7 ...
<Program>
-<Block>
--[{}
--<StatementList>
---<Statement>
----<VarDecl>
-----<Type>
-----[int]
-----<Id>
-----[a]
---<StatementList>
----<Statement>

```



```

-----<AssignStatement>
-----<Id>
-----[a]
-----[=]
-----<Expression>
-----<IntegerExpression>
-----<Digit>
-----[3]
-----<IntOp>
-----[+]
-----<Expression>
-----<IntegerExpression>
-----<Digit>
-----[2]
--[}]
-[$]

```

```

SEMANTIC ANALYSIS: Beginning Semantic Analysis on Program 7 ...
SEMANTIC ANALYSIS: New Scope [ 0 ] has been entered at line: 14.
SEMANTIC ANALYSIS: Variable [ a ] has been declared at (14:2)
SEMANTIC ANALYSIS: Variable [ a ] has been initialized at (15:9)
SEMANTIC ANALYSIS: WARNING: Variable [ a ] is declared and initialized but never
used.

```

Program 7 Semantic Analysis produced 0 error(s) and 1 warning(s).

AST for program 7 ...

```

<BLOCK>
-<VariableDeclaration>
--[int]
--[a]
-<Assign>
--[a]
--<Addition>
---[3]
---[2]

```

Program 7 Symbol Table

Name	Type	Scope	Line
a	int	0	14

The first two programs complete Lex, Parse, and Semantic Analysis without any errors. Since there are no variables in either, after passing parse, there isn't any scope/type checking to do. The third program fails due to a parser error of an extra closing brace, so semantic analysis is skipped and the AST and symbol table are also skipped. The fourth program fails from lex because of the illegal symbol, so it also does not continue to complete parse or semantic analysis. The fifth program fails at semantic analysis because we are trying to set $a = b$, and a is of type int and b is of type string, it also throws warnings for the variables if they are unused or uninitialized. The sixth program fails at parse because after type, an id is expected and instead we got another type. The seventh program passes semantic analysis, lex, and parse, since there are no type errors or scope errors. It prints out the AST and symbol table for the program.

BOOLEAN TEST CASES (NON-VERBOSE MODE)

Input text:

```
{int i while(1!=(1==(1!=(1!=a)))){}}$  
  
{boolean s while(s!=(s==(s!=(1!=2)))){}}$  
  
{string a boolean b if (false == (b == (a != "string"))) {}}$  
  
{boolean b if(b==(b!=b)){}$  
  
{string s int i if(i==s){}}$  
  
{int a a = 1 + (3 != 3)}$  
  
{int a print(1 + (3 != 3))}$  
  
{int a a=1+3}$  
  
{print(("aa" != 1))}$  
  
{boolean a a = (true == (a != (true != true)))}$
```

Output:

```
INFO  Lexer - Lexing program 1...  
INFO  Lexer - Lex completed with 0 errors
```

```
PARSER: Parsing program 1 ...  
PARSER: Parse completed successfully
```

```
CST for program 1 ...  
<Program>  
-<Block>  
--[{}]  
--<StatementList>  
---<Statement>  
----<VarDecl>  
-----<Type>  
-----[int]  
-----<Id>  
-----[i]  
---<StatementList>  
----<Statement>  
-----<WhileStatement>  
-----[while]  
-----<BooleanExpression>  
-----[()]  
-----<Expression>  
-----<IntegerExpression>  
-----<Digit>  
-----[1]  
-----<BoolOp>  
-----[!=]  
-----<Expression>
```

```

-----<BooleanExpression>
-----[ ( ]
-----<Expression>
-----<IntegerExpression>
-----<Digit>
-----[ 1 ]
-----<BoolOp>
-----[ == ]
-----<Expression>
-----<BooleanExpression>
-----[ ( ]
-----<Expression>
-----<IntegerExpression>
-----<Digit>
-----[ 1 ]
-----<BoolOp>
-----[ != ]
-----<Expression>
-----<BooleanExpression>
-----[ ( ]
-----<Expression>
-----<IntegerExpression>
-----<Digit>
-----[ 1 ]
-----<BoolOp>
-----[ != ]
-----<Expression>
-----<Id>
-----[ a ]
-----[ ) ]
-----[ ] ]
-----[ ] ]
-----[ ] ]
-----[ ] ]
-----<Block>
-----[ { ]
-----[ StatementList ]
-----[ } ]
-- [ ] ]
- [ $ ]

```

SEMANTIC ANALYSIS: Beginning Semantic Analysis on Program 1 ...

SEMANTIC ANALYSIS: ERROR: Undeclared variable [a] was used at (1:29) before being declared.

SEMANTIC ANALYSIS: ERROR: Incorrect Type Comparison - Variable [1] of type [int] was compared to type [BooleanExpression] at (1:30).

SEMANTIC ANALYSIS: ERROR: Incorrect Type Comparison - Variable [1] of type [int] was compared to type [BooleanExpression] at (1:31).

SEMANTIC ANALYSIS: ERROR: Incorrect Type Comparison - Variable [1] of type [int] was compared to type [BooleanExpression] at (1:32).

SEMANTIC ANALYSIS: WARNING: Variable [i] is declared but never initialized or used.

Program 1 Semantic Analysis produced 4 error(s) and 1 warning(s).

AST for program 1: Skipped due to SEMANTIC ANALYSIS error(s)

Symbol Table for program 1: Skipped due to SEMANTIC ANALYSIS error(s)

INFO Lexer - Lexing program 2...
INFO Lexer - Lex completed with 0 errors

PARSER: Parsing program 2 ...
PARSER: Parse completed successfully

CST for program 2 ...

```
<Program>
-<Block>
--[{}
--<StatementList>
---<Statement>
----<VarDecl>
-----<Type>
-----[boolean]
-----<Id>
-----[s]
---<StatementList>
----<Statement>
-----<WhileStatement>
-----[while]
-----<BooleanExpression>
-----[()]
-----<Expression>
-----<Id>
-----[s]
-----<BoolOp>
-----[!=]
-----<Expression>
-----<BooleanExpression>
-----[()]
-----<Expression>
-----<Id>
-----[s]
-----<BoolOp>
-----[==]
-----<Expression>
-----<BooleanExpression>
-----[()]
-----<Expression>
-----<Id>
-----[s]
-----<BoolOp>
-----[!=]
-----<Expression>
-----<BooleanExpression>
-----[()]
-----<Expression>
-----<IntegerExpression>
-----<Digit>
-----[1]
-----<BoolOp>
-----[!=]
```

```

-----<Expression>
-----<IntegerExpression>
-----<Digit>
-----[2]
-----[)]
-----[)]
-----[)]
-----[)]
-----<Block>
-----[{}
-----[StatementList]
-----[}]
--[}]
-[$]

```

SEMANTIC ANALYSIS: Beginning Semantic Analysis on Program 2 ...
SEMANTIC ANALYSIS: WARNING: Variable [s] is declared and used but never initialized.

Program 2 Semantic Analysis produced 0 error(s) and 1 warning(s).

AST for program 2 ...

```

<BLOCK>
-<VariableDeclaration>
--[boolean]
--[s]
-<While>
--<isNotEqual>
---[s]
---<isEqual>
----[s]
----<isNotEqual>
-----[s]
-----<isNotEqual>
-----[1]
-----[2]
--[BLOCK]

```

Program 2 Symbol Table

```

-----
Name  Type      Scope  Line
-----
s      boolean    0       3

```

INFO Lexer - Lexing program 3...
INFO Lexer - Lex completed with 0 errors

PARSER: Parsing program 3 ...
PARSER: Parse completed successfully

CST for program 3 ...

```

<Program>
-<Block>
--[{}
--<StatementList>

```

```

---<Statement>
----<VarDecl>
-----<Type>
-----[string]
-----<Id>
-----[a]
---<StatementList>
----<Statement>
-----<VarDecl>
-----<Type>
-----[boolean]
-----<Id>
-----[b]
---<StatementList>
----<Statement>
-----<IfStatement>
-----[if]
-----<BooleanExpression>
-----[()]
-----<Expression>
-----<BooleanExpression>
-----<BoolVal>
-----[false]
-----<BoolOp>
-----[==]
-----<Expression>
-----<BooleanExpression>
-----[()]
-----<Expression>
-----<Id>
-----[b]
-----<BoolOp>
-----[==]
-----<Expression>
-----<BooleanExpression>
-----[()]
-----<Expression>
-----<Id>
-----[a]
-----<BoolOp>
-----[!=]
-----<Expression>
-----<StringExpression>
-----[""]
-----<CharList>
-----<Char>
-----[s]
-----<CharList>
-----<Char>
-----[t]
-----<CharList>
-----<Char>
-----[r]
-----<CharList>
-----<Char>
-----[i]

```

```

-----<CharList>
-----<Char>
-----[n]
-----<CharList>
-----<Char>
-----[g]
-----["]
-----[]]
-----[]]
-----[]]
-----<Block>
-----[{}
-----[StatementList]
-----[]]
--[]]
-[$]

```

SEMANTIC ANALYSIS: Beginning Semantic Analysis on Program 3 ...
SEMANTIC ANALYSIS: WARNING: Variable [b] is declared and used but never
initialized.

Program 3 Semantic Analysis produced 0 error(s) and 1 warning(s).

AST for program 3 ...

```

<BLOCK>
-<VariableDeclaration>
--[string]
--[a]
-<VariableDeclaration>
--[boolean]
--[b]
-<If>
--<isEqual>
---[false]
---<isEqual>
----[b]
----<isNotEqual>
-----[a]
-----["string"]
--[BLOCK]

```

Program 3 Symbol Table

Name	Type	Scope	Line
b	boolean	0	5
a	string	0	5

INFO Lexer - Lexing program 4...
INFO Lexer - Lex completed with 0 errors

PARSER: Parsing program 4 ...
PARSER: Parse completed successfully

CST for program 4 ...

```

<Program>
-<Block>
--[{}
--<StatementList>
---<Statement>
----<VarDecl>
-----<Type>
-----[boolean]
-----<Id>
-----[b]
---<StatementList>
----<Statement>
-----<IfStatement>
-----[if]
-----<BooleanExpression>
-----[()]
-----<Expression>
-----<Id>
-----[b]
-----<BoolOp>
-----[==]
-----<Expression>
-----<BooleanExpression>
-----[()]
-----<Expression>
-----<Id>
-----[b]
-----<BoolOp>
-----[!=]
-----<Expression>
-----<Id>
-----[b]
-----[)]
-----[)]
-----<Block>
-----[{}
-----[StatementList]
-----[}]
--[}]
-[$]

```

SEMANTIC ANALYSIS: Beginning Semantic Analysis on Program 4 ...

SEMANTIC ANALYSIS: WARNING: Variable [b] is declared and used but never initialized.

Program 4 Semantic Analysis produced 0 error(s) and 1 warning(s).

AST for program 4 ...

```

<BLOCK>
-<VariableDeclaration>
--[boolean]
--[b]
-<If>
--<isEqual>
---[b]
---<isNotEqual>

```



```

----[b]
----[b]
--[BLOCK]

```

Program 4 Symbol Table

```

-----
Name   Type      Scope  Line
-----
b      boolean   0      7

```

```

INFO  Lexer - Lexing program 5...
INFO  Lexer - Lex completed with 0 errors

```

```

PARSER: Parsing program 5 ...
PARSER: Parse completed successfully

```

CST for program 5 ...

```

<Program>
-<Block>
--[{}
--<StatementList>
---<Statement>
----<VarDecl>
-----<Type>
-----[string]
-----<Id>
-----[s]
---<StatementList>
----<Statement>
-----<VarDecl>
-----<Type>
-----[int]
-----<Id>
-----[i]
----<StatementList>
-----<Statement>
-----<IfStatement>
-----[if]
-----<BooleanExpression>
-----[()]
-----<Expression>
-----<Id>
-----[i]
-----<BoolOp>
-----[==]
-----<Expression>
-----<Id>
-----[s]
-----[]
-----<Block>
-----[{}
-----[StatementList]
-----[]
--[]
-[$]

```

```

SEMANTIC ANALYSIS: Beginning Semantic Analysis on Program 5 ...
SEMANTIC ANALYSIS: ERROR: Incorrect Type Comparison - Variable [ i ] of type [ int
] was compared to type [ string ] at (9:23).
SEMANTIC ANALYSIS: WARNING: Variable [ i ] is declared and used but never
initialized.
SEMANTIC ANALYSIS: WARNING: Variable [ s ] is declared and used but never
initialized.

```

Program 5 Semantic Analysis produced 1 error(s) and 2 warning(s).

AST for program 5: Skipped due to SEMANTIC ANALYSIS error(s)

Symbol Table for program 5: Skipped due to SEMANTIC ANALYSIS error(s)

```

INFO  Lexer - Lexing program 6...
INFO  Lexer - Lex completed with 0 errors

```

```

PARSER: Parsing program 6 ...
PARSER: Parse completed successfully

```

```

CST for program 6 ...
<Program>
-<Block>
--[{}
--<StatementList>
---<Statement>
----<VarDecl>
-----<Type>
-----[int]
-----<Id>
-----[a]
---<StatementList>
----<Statement>
-----<AssignStatement>
-----<Id>
-----[a]
-----[=]
-----<Expression>
-----<IntegerExpression>
-----<Digit>
-----[1]
-----<IntOp>
-----[+]
-----<Expression>
-----<BooleanExpression>
-----[()]
-----<Expression>
-----<IntegerExpression>
-----<Digit>
-----[3]
-----<BoolOp>
-----[!=]
-----<Expression>
-----<IntegerExpression>

```

```

-----<Digit>
-----[3]
-----[]
--[{}]
-[$]

```

```

SEMANTIC ANALYSIS: Beginning Semantic Analysis on Program 6 ...
SEMANTIC ANALYSIS: ERROR: Type Mismatch - [ IntOp ] of type [ int ] was assigned to
type [ BooleanExpression ] at (11:23).
SEMANTIC ANALYSIS: WARNING: Variable [ a ] is declared and initialized but never
used.

```

Program 6 Semantic Analysis produced 1 error(s) and 1 warning(s).

AST for program 6: Skipped due to SEMANTIC ANALYSIS error(s)

Symbol Table for program 6: Skipped due to SEMANTIC ANALYSIS error(s)

```

INFO  Lexer - Lexing program 7...
INFO  Lexer - Lex completed with 0 errors

```

```

PARSER: Parsing program 7 ...
PARSER: Parse completed successfully

```

```

CST for program 7 ...
<Program>
-<Block>
--[{}
--<StatementList>
---<Statement>
----<VarDecl>
-----<Type>
-----[int]
-----<Id>
-----[a]
---<StatementList>
----<Statement>
-----<PrintStatement>
-----[print]
-----[()]
-----<Expression>
-----<IntegerExpression>
-----<Digit>
-----[1]
-----<IntOp>
-----[+]
-----<Expression>
-----<BooleanExpression>
-----[()]
-----<Expression>
-----<IntegerExpression>
-----<Digit>
-----[3]
-----<BoolOp>
-----[!=]

```

```

-----<Expression>
-----<IntegerExpression>
-----<Digit>
-----[3]
-----[]
-----[]
--[]
-[$]

```

```

SEMANTIC ANALYSIS: Beginning Semantic Analysis on Program 7 ...
SEMANTIC ANALYSIS: ERROR: Type Mismatch - [ IntOp ] of type [ int ] was assigned to
type [ BooleanExpression ] at (13:25).
SEMANTIC ANALYSIS: WARNING: Variable [ a ] is declared but never initialized or
used.

```

Program 7 Semantic Analysis produced 1 error(s) and 1 warning(s).

AST for program 7: Skipped due to SEMANTIC ANALYSIS error(s)

Symbol Table for program 7: Skipped due to SEMANTIC ANALYSIS error(s)

```

INFO  Lexer - Lexing program 8...
INFO  Lexer - Lex completed with 0 errors

```

```

PARSER: Parsing program 8 ...
PARSER: Parse completed successfully

```

```

CST for program 8 ...
<Program>
-<Block>
--[{}
--<StatementList>
---<Statement>
----<VarDecl>
-----<Type>
-----[int]
-----<Id>
-----[a]
---<StatementList>
----<Statement>
-----<AssignStatement>
-----<Id>
-----[a]
-----[=]
-----<Expression>
-----<IntegerExpression>
-----<Digit>
-----[1]
-----<IntOp>
-----[+]
-----<Expression>
-----<IntegerExpression>
-----<Digit>
-----[3]
--[]

```

-\$]

SEMANTIC ANALYSIS: Beginning Semantic Analysis on Program 8 ...
SEMANTIC ANALYSIS: WARNING: Variable [a] is declared and initialized but never used.

Program 8 Semantic Analysis produced 0 error(s) and 1 warning(s).

AST for program 8 ...

<BLOCK>

-<VariableDeclaration>

--[int]

--[a]

-<Assign>

--[a]

--<Addition>

---[1]

---[3]

Program 8 Symbol Table

Name	Type	Scope	Line
a	int	0	15

INFO Lexer - Lexing program 9...

INFO Lexer - Lex completed with 0 errors

PARSER: Parsing program 9 ...

PARSER: Parse completed successfully

CST for program 9 ...

<Program>

-<Block>

--[{}]

--<StatementList>

---<Statement>

----<PrintStatement>

-----[print]

-----[()]

-----<Expression>

-----<BooleanExpression>

-----[()]

-----<Expression>

-----<StringExpression>

-----["]

-----<CharList>

-----<Char>

-----[a]

-----<CharList>

-----<Char>

-----[a]

-----["]

-----<BoolOp>

-----[!=]

```

-----<Expression>
-----<IntegerExpression>
-----<Digit>
-----[1]
-----[]]
-----[]]
--[]]
-[$]

```

SEMANTIC ANALYSIS: Beginning Semantic Analysis on Program 9 ...
 SEMANTIC ANALYSIS: ERROR: Incorrect Type Comparison - Variable ["aa"] of type [string] was compared to type [int] at (17:17).

Program 9 Semantic Analysis produced 1 error(s) and 0 warning(s).

AST for program 9: Skipped due to SEMANTIC ANALYSIS error(s)

Symbol Table for program 9: Skipped due to SEMANTIC ANALYSIS error(s)

INFO Lexer - Lexing program 10...
 INFO Lexer - Lex completed with 0 errors

PARSER: Parsing program 10 ...
 PARSER: Parse completed successfully

CST for program 10 ...

```

<Program>
-<Block>
--[{}
--<StatementList>
---<Statement>
----<VarDecl>
-----<Type>
-----[boolean]
-----<Id>
-----[a]
---<StatementList>
----<Statement>
-----<AssignStatement>
-----<Id>
-----[a]
-----[=]
-----<Expression>
-----<BooleanExpression>
-----[()]
-----<Expression>
-----<BooleanExpression>
-----<BoolVal>
-----[true]
-----<BoolOp>
-----[==]
-----<Expression>
-----<BooleanExpression>
-----[()]
-----<Expression>

```

```

-----<Id>
-----[a]
-----<BoolOp>
-----[!=]
-----<Expression>
-----<BooleanExpression>
-----[(]
-----<Expression>
-----<BooleanExpression>
-----<BoolVal>
-----[true]
-----<BoolOp>
-----[!=]
-----<Expression>
-----<BooleanExpression>
-----<BoolVal>
-----[true]
-----[)]
-----[)]
-----[)]
--[}]
-[$]

```

SEMANTIC ANALYSIS: Beginning Semantic Analysis on Program 10 ...

Program 10 Semantic Analysis produced 0 error(s) and 0 warning(s).

AST for program 10 ...

```

<BLOCK>
-<VariableDeclaration>
--[boolean]
--[a]
-<Assign>
--[a]
--<isEqual>
---[true]
---<isNotEqual>
----[a]
----<isNotEqual>
-----[true]
-----[true]

```

Program 10 Symbol Table

Name	Type	Scope	Line
a	boolean	0	19

The first program fails semantic analysis for multiple errors. The variable "a" in (1 != a) is undeclared, also we are trying to compare BooleanExpressions to the integer "1", so it throws multiple errors for that. The second program passes, because (1!=2) will produce a boolVal (false) and then we are comparing that to the boolean variable "s" over and over, so there are no type errors. The third and fourth programs also pass, because all variables are defined before they are used, and there are no type mismatch errors. The fifth program fails because we are trying to compare string s to int i; two variables of different types. The sixth and seventh programs fail because we are trying to add 1 + BooleanExpression in both a print statement and an assign statement. The eighth program passes since a is an int and we are setting it equal to an intOp.

The ninth program fails due to type mismatch comparing a string to an int within a print statement. The tenth program is an extreme boolean expression that passes since all variables used in the expression are of type boolean.

SCOPE TEST CASES (NON-VERBOSE MODE)

Input text:

```
{boolean a {{{{{{print(1+1)}}}}}}}$
```

```
{string a a=1}$
```

```
{boolean b b = "true"}$
```

```
{int a {boolean h {a =4} h = true}}$
```

Output:

```
INFO  Lexer - Lexing program 1...
INFO  Lexer - Lex completed with 0 errors
```

```
PARSER: Parsing program 1 ...
PARSER: Parse completed successfully
```

CST for program 1 ...

```
<Program>
-<Block>
--[{}]
--<StatementList>
---<Statement>
----<VarDecl>
-----<Type>
-----[boolean]
-----<Id>
-----[a]
---<StatementList>
----<Statement>
-----<Block>
-----[{}]
-----<StatementList>
-----<Statement>
-----<Block>
-----[{}]
-----<StatementList>
-----<Statement>
-----<Block>
-----[{}]
-----<StatementList>
-----<Statement>
-----<Block>
-----[{}]
-----<StatementList>
-----<Statement>
-----<Block>
-----[{}]
```



```

-----<StatementList>
-----<Statement>
-----<Block>
-----[{}
-----<StatementList>
-----<Statement>
-----<Block>
-----[{}
-----<StatementList>
-----<Statement>
-----<Block>
-----[{}
-----<StatementList>
-----<Statement>
-----<PrintStatement>
-----[print]
-----[(]
-----<Expression>
-----<IntegerExpression>
-----<Digit>
-----[1]
-----<IntOp>
-----[+]
-----<Expression>
-----<IntegerExpression>
-----<Digit>
-----[1]
-----[])
-----[]
-----[]}
-----}]
-----}]]
-----[]
-----[]
-----[]
-----[]
-----[]
-----[]
-----[]
-----[]
-----[]

```

```

SEMANTIC ANALYSIS: Beginning Semantic Analysis on Program 1 ...
SEMANTIC ANALYSIS: WARNING: Variable [ a ] is declared but never initialized or
used.

```

Program 1 Semantic Analysis produced 0 error(s) and 1 warning(s).

AST for program 1 ...

```
<BLOCK>
-<VariableDeclaration>
--[boolean]
--[a]
-<BLOCK>
--<BLOCK>
---<BLOCK>
----<BLOCK>
-----<BLOCK>
-----<BLOCK>
```

```

-----<BLOCK>
-----<BLOCK>
-----<Print>
-----<Addition>
-----[1]
-----[1]

```

Program 1 Symbol Table

```

-----
Name  Type      Scope  Line
-----
a      boolean    0      1

```

```

INFO  Lexer - Lexing program 2...
INFO  Lexer - Lex completed with 0 errors

```

```

PARSER: Parsing program 2 ...
PARSER: Parse completed successfully

```

CST for program 2 ...

```

<Program>
-<Block>
--[{}
--<StatementList>
---<Statement>
----<VarDecl>
-----<Type>
-----[string]
-----<Id>
-----[a]
---<StatementList>
----<Statement>
-----<AssignStatement>
-----<Id>
-----[a]
-----[=]
-----<Expression>
-----<IntegerExpression>
-----<Digit>
-----[1]
--[]
-[$]

```

```

SEMANTIC ANALYSIS: Beginning Semantic Analysis on Program 2 ...
SEMANTIC ANALYSIS: ERROR: Type Mismatch - Variable [ a ] of type [ string ] was
assigned to type [ int ] at (3:13).
SEMANTIC ANALYSIS: WARNING: Variable [ a ] is declared but never initialized or
used.

```

Program 2 Semantic Analysis produced 1 error(s) and 1 warning(s).

AST for program 2: Skipped due to SEMANTIC ANALYSIS error(s)

Symbol Table for program 2: Skipped due to SEMANTIC ANALYSIS error(s)

INFO Lexer - Lexing program 3...
INFO Lexer - Lex completed with 0 errors

PARSER: Parsing program 3 ...
PARSER: Parse completed successfully

CST for program 3 ...

```
<Program>
-<Block>
--[{}
--<StatementList>
---<Statement>
----<VarDecl>
-----<Type>
-----[boolean]
-----<Id>
-----[b]
---<StatementList>
----<Statement>
-----<AssignStatement>
-----<Id>
-----[b]
-----[=]
-----<Expression>
-----<StringExpression>
-----["]
-----<CharList>
-----<Char>
-----[t]
-----<CharList>
-----<Char>
-----[r]
-----<CharList>
-----<Char>
-----[u]
-----<CharList>
-----<Char>
-----[e]
-----["]
--[]
-[$]
```

SEMANTIC ANALYSIS: Beginning Semantic Analysis on Program 3 ...
SEMANTIC ANALYSIS: ERROR: Type Mismatch - Variable [b] of type [boolean] was assigned to type [string] at (5:16).
SEMANTIC ANALYSIS: WARNING: Variable [b] is declared but never initialized or used.

Program 3 Semantic Analysis produced 1 error(s) and 1 warning(s).

AST for program 3: Skipped due to SEMANTIC ANALYSIS error(s)

Symbol Table for program 3: Skipped due to SEMANTIC ANALYSIS error(s)

```
INFO  Lexer - Lexing program 4...
INFO  Lexer - Lex completed with 0 errors
```

```
PARSER: Parsing program 4 ...
PARSER: Parse completed successfully
```

```
CST for program 4 ...
```

```
<Program>
-<Block>
--[{}
--<StatementList>
---<Statement>
----<VarDecl>
-----<Type>
-----[int]
-----<Id>
-----[a]
---<StatementList>
----<Statement>
-----<Block>
-----[{}
-----<StatementList>
-----<Statement>
-----<VarDecl>
-----<Type>
-----[boolean]
-----<Id>
-----[h]
-----<StatementList>
-----<Statement>
-----<Block>
-----[{}
-----<StatementList>
-----<Statement>
-----<AssignStatement>
-----<Id>
-----[a]
-----[=]
-----<Expression>
-----<IntegerExpression>
-----<Digit>
-----[4]
-----[}]
-----<StatementList>
-----<Statement>
-----<AssignStatement>
-----<Id>
-----[h]
-----[=]
-----<Expression>
-----<BooleanExpression>
-----<BoolVal>
-----[true]
-----[}]
--[{}
-[$]
```

```

SEMANTIC ANALYSIS: Beginning Semantic Analysis on Program 4 ...
SEMANTIC ANALYSIS: WARNING: Variable [ a ] is declared and initialized but never
used.
SEMANTIC ANALYSIS: WARNING: Variable [ h ] is declared and initialized but never
used.

```

Program 4 Semantic Analysis produced 0 error(s) and 2 warning(s).

```

AST for program 4 ...
<BLOCK>
-<VariableDeclaration>
--[int]
--[a]
-<BLOCK>
--<VariableDeclaration>
---[boolean]
---[h]
--<BLOCK>
---<Assign>
----[a]
----[4]
--<Assign>
---[h]
---[true]

```

Program 4 Symbol Table

Name	Type	Scope	Line
a	int	0	7
h	boolean	1	7

The first program passes, but just throws a warning for variable a saying that it is declared but never initialized or used. The second program fails because we are trying to assign string a a value of '1'. Program 3 also fails because we are trying to assign a boolean a string value. The fourth program passes since both a and h are declared before they are used.

SUPER LONG TEST CASE (NON-VERBOSE MODE)

Input text:

```

{
    int a
    a = 0
    string z
    z = "bond"
    while (a != 9) {
        if (a != 5) {
            print("bond")
        }
    }
    {
        a = 1 + a
        string b
    }
}

```

```

        b = "james bond"
        print(b)
    }
}
{}
boolean c
c = true
boolean d
d = (true == (true == false))
d = (a == b)
d = (1 == a)
d = (1 != 1)
d = ("string" == 1)
d = (a != "string")
d = ("string" != "string")
if (d == true) {
    int c
    c = 1 + d
    if (c == 1) {
        print("ugh")
    }
}
while ("string" == a) {
    while (1 == true) {
        a = 1 + "string"
    }
}
}
}$

```

Output:

```

INFO  Lexer - Lexing program 1...
INFO  Lexer - Lex completed with 0 errors

```

```

PARSER: Parsing program 1 ...
PARSER: Parse completed successfully

```

```

CST for program 1 ...
(NOT SHOWN DUE TO EXTREME LENGTH - But no errors thrown, so will display if running
code in terminal)

```

```

SEMANTIC ANALYSIS: Beginning Semantic Analysis on Program 1 ...
SEMANTIC ANALYSIS: ERROR: Undeclared variable [ b ] was used at (22:16) before
being declared.
SEMANTIC ANALYSIS: ERROR: Incorrect Type Comparison - Variable [ "string" ] of type
[ string ] was compared to type [ int ] at (25:23).
SEMANTIC ANALYSIS: ERROR: Incorrect Type Comparison - Variable [ a ] of type [ int
] was compared to type [ string ] at (26:16).
SEMANTIC ANALYSIS: ERROR: Type Mismatch - [ IntOp ] of type [ int ] was assigned to
type [ boolean ] at (30:18).
SEMANTIC ANALYSIS: ERROR: Incorrect Type Comparison - Variable [ "string" ] of type
[ string ] was compared to type [ int ] at (35:25).
SEMANTIC ANALYSIS: ERROR: Incorrect Type Comparison - Variable [ 1 ] of type [ ]
was compared to type [ boolean ] at (36:22).
SEMANTIC ANALYSIS: ERROR: Type Mismatch - [ IntOp ] of type [ int ] was assigned to

```

```
type [ string ] at (37:29).  
SEMANTIC ANALYSIS: WARNING: Variable [ z ] is declared and initialized but never  
used.  
SEMANTIC ANALYSIS: WARNING: Variable [ c ] is declared and initialized but never  
used.
```

Program 1 Semantic Analysis produced 7 error(s) and 2 warning(s).

AST for program 1: Skipped due to SEMANTIC ANALYSIS error(s)

Symbol Table for program 1: Skipped due to SEMANTIC ANALYSIS error(s)

This test case fails semantic analysis for many different reasons. The first is because variable "b" is used outside of its scope in the expression $d = (a == b)$. Next, there are some type comparison errors trying to compare "string" to 1 and "string" to variable a (type int). Then, we are trying to add $1+d$ (d is of type boolean). Then, again int a is compared to "string" and 1 is compared to the boolVal true. Finally, we try to add $1 + \text{"string"}$ which is invalid, so it throws another error.