System Test Plan
Graph-Based Coverage

By: Abdullah Sakallah
Mohammed Zakiuddin

Table of Contents

1. System Under Test

Compilers is a free and open-source project accessible on GitHub. The project under test covers a compiler that parses the input and generates a syntax tree. The project was created by developing context-free grammar in order to generate parsing rules and perform lexical analysis on the language.

1.1. Details

Project source code: https://github.com/Niljas/Compiler-Design-Project

Number of the components for Compilers

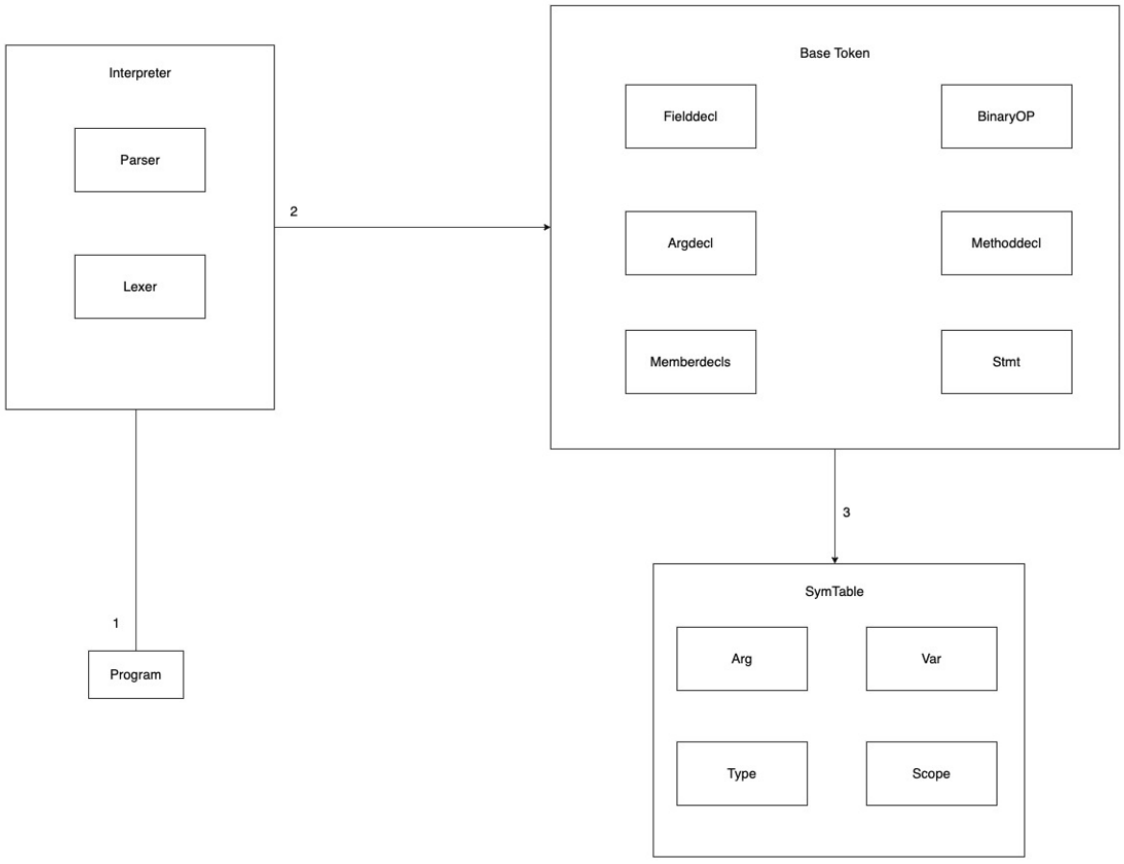
- Compiler Design Project (CDP):
  - 2 packages
  - 15 class files
  - 1 interface
  - 1 make file

Lines of code for each component

- Argdecl: 17 LOC
- BaseToken: 134 LOC
- BinaryOP: 19 LOC
- Checkable: 6 LOC (Interface)
- Compiler-Design-Project-Main: 12 LOC (Package 1)
- Expr: 207 LOC
- Fielddecl: 46 LOC
- Grammar: 97 LOC
- Interpreter: 37 LOC
- LexerRules: 17 LOC
- Makefile: 75 LOC (Make File)
- Memberdecls: 38 LOC
- Methoddecl: 66 LOC
- Name: 37 LOC
- ParserTest: 30 LOC
- Program: 16 LOC
- Stmt: 215 LOC
- SymTable: 74 LOC
- Tokens: 314 LOC (Package 2)
- TypeChecking: 32 LOC

Total LOC: 1489

1.2. Software Architecture

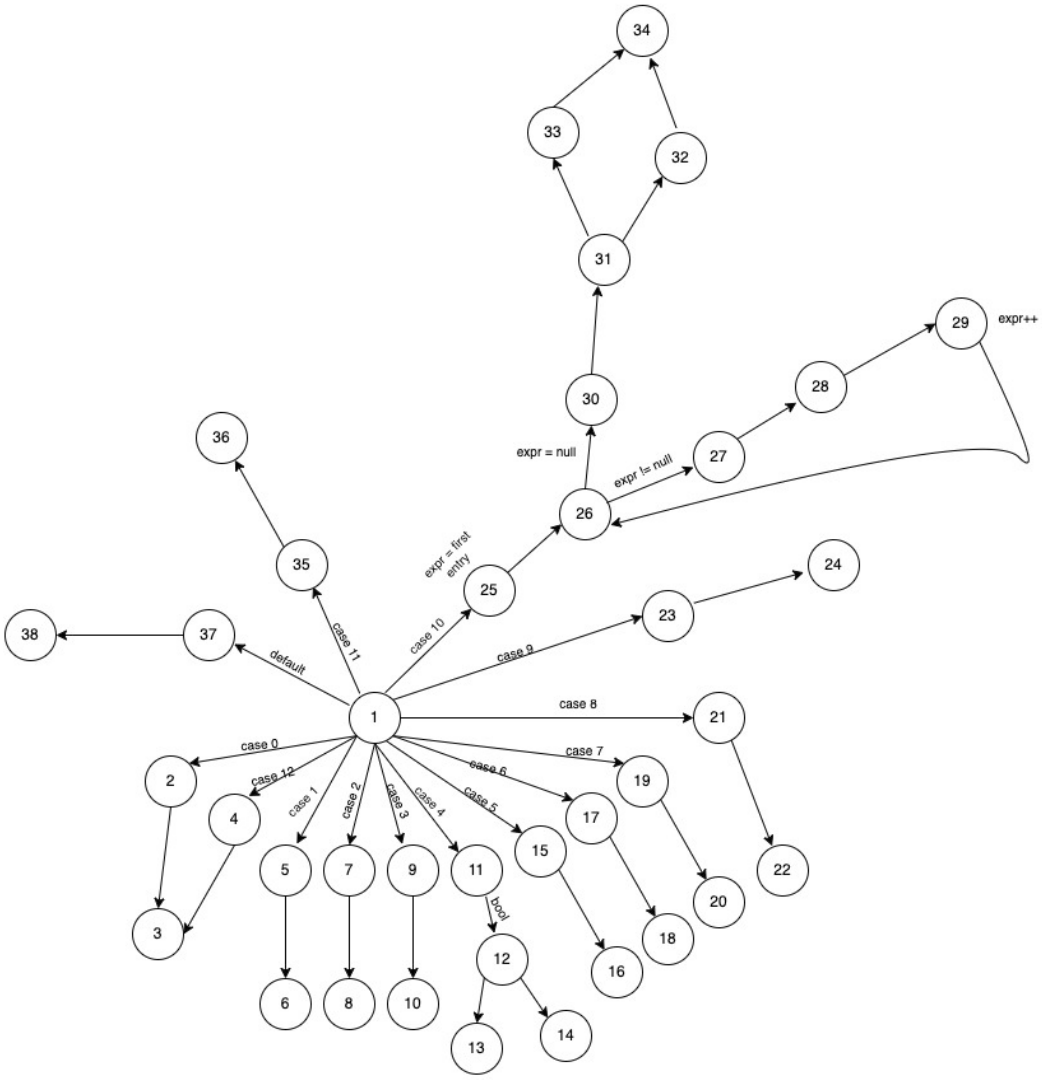


2. Test Environment

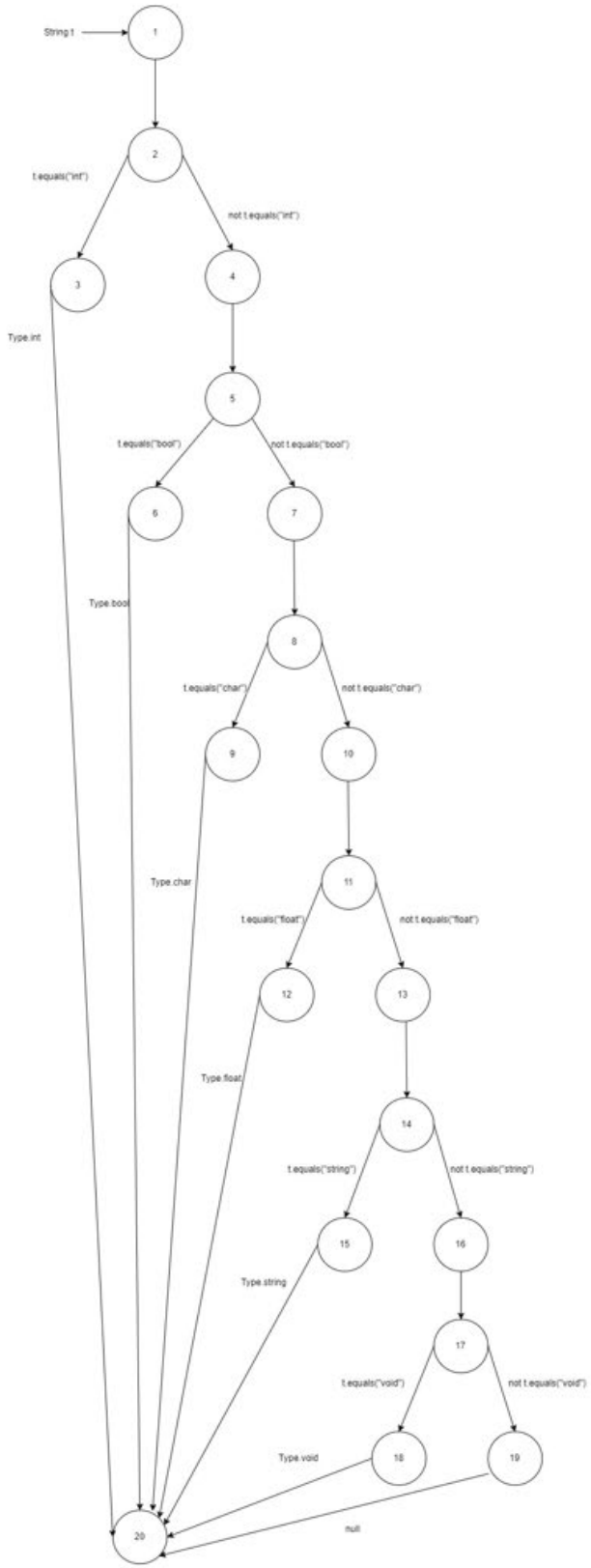
- The project under test is deployed under IntelliJ IDEA 2022.1.1 (Community Edition).
- It uses Junit 5 for unit testing.
- The Operating System that is used to run the tests is Windows 11 & Mac OS (Monterey 12.6).
- The version of Java that is being used is 17.0.2.

3. Control Flow

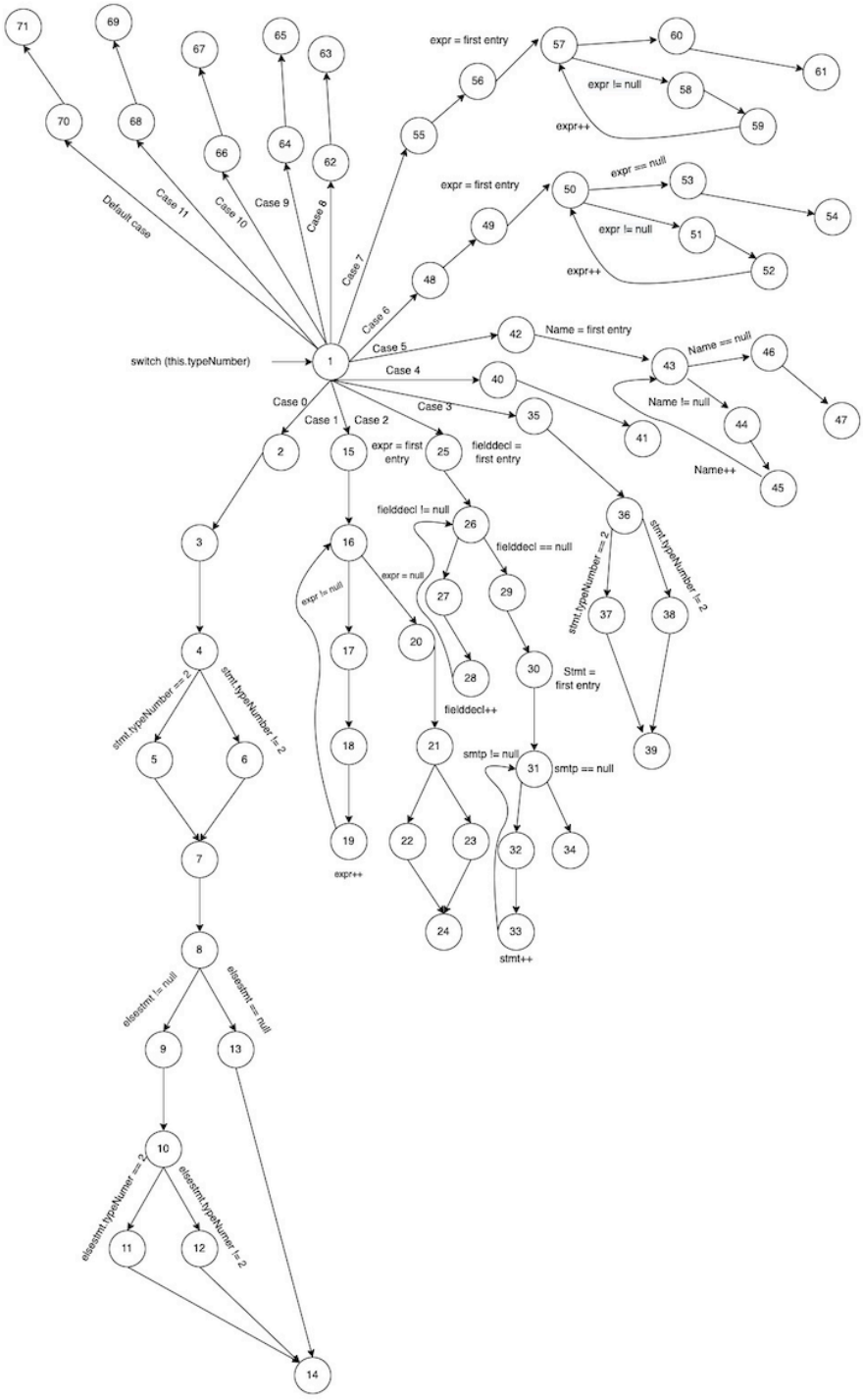
3.1. [expr] [toString()] Control Flow Graph



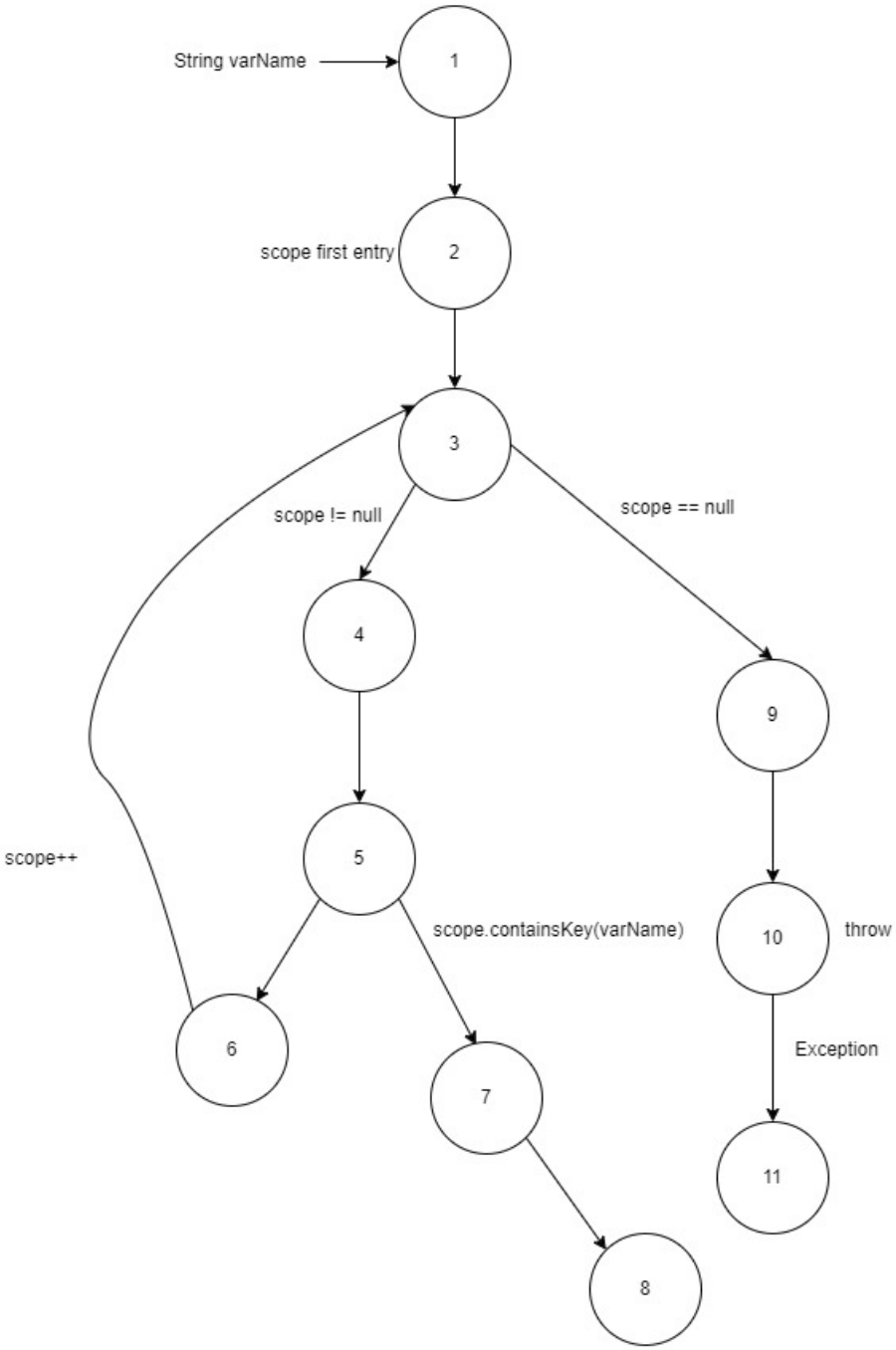
3.2 [BaseToken] [getType(String t)] Control Flow Graph



3.3 [Stmt] [toString()] Control Flow Graph



3.4 [SymTable][getVar(String varName)]



4. Test Requirements

Prime Paths:

PPC: TR contains each prime path in graph 3.1 [expr][toString()] & 3.3 [stmt][toString()]

Prime Paths (P1) = { [1, 2, 3], [1, 4, 3], [1, 5, 6], [1, 7, 8], [1, 9, 10], [1, 11, 12, 13], [1, 11, 12,14], [1, 15, 16], [1, 17, 18], [1, 19, 20], [1, 21, 22], [1, 23, 24], [1, 25, 26, 30, 31, 32, 34], [1, 25, 26, 30, 31, 33, 34], [1, 25, 26, 27, 28, 29], [1, 35, 36], [1, 37, 38], [26, 27, 28, 29, 26], [27, 28, 29, 26, 27, 28, 29], [27, 28, 29, 26, 30, 31, 32, 34], [27, 28, 29, 26, 30, 31, 33, 34]}

TR = 21 Prime Paths

Prime Paths (P2) = { [1, 2, 3, 4, 5, 7, 8, 9, 10, 11, 14], [1, 2, 3, 4, 5, 7, 8, 9, 10, 12, 14], [1, 2, 3, 4, 5, 7, 8, 13, 14] [1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 14], [1, 2, 3, 4, 6, 7, 8, 9, 10, 12, 14], [1, 2, 3, 4, 6, 7, 8, 13, 14], [1, 15, 16, 17, 18, 19], [1, 15, 16, 20, 21, 22, 24], [1, 15, 16, 20, 21, 23, 24], [16, 17, 18, 19, 16], [17, 18, 19 ,16, 17, 18, 19], [17, 18, 19 ,16, 20, 21, 22, 24], [17, 18, 19 ,16, 20, 21, 23, 24], [1, 25, 26, 27, 28], [1, 25, 26, 29, 30, 31, 34], [1, 25, 26, 29, 30, 31, 32, 33], [26, 27, 28, 26], [27, 28, 26, 27, 28], [27, 28, 26, 29, 30, 31, 34], [27, 28, 26, 29, 30, 31, 32, 33], [31, 32, 33, 31], [32, 33, 31, 32, 33], [32, 33, 31, 34], [1, 35, 36, 37, 39], [1, 35, 36, 38, 39], [1,40,41], [43, 44, 45, 43], [44, 45, 43, 44, 45], [44, 45, 43, 46, 47], [1,42,43,46,47], [1,42,43,44,45], [50, 51, 52, 50], [51, 52, 50, 51, 52], [51, 52, 50, 53, 54], [1, 48, 49, 50, 53, 54], [1, 48, 49, 50, 51, 52], [1, 55, 56, 57, 60, 61], [1, 55, 56, 57, 58, 59], [57, 58, 59, 57], [58, 59 , 57, 58, 59], [58, 59, 57, 60, 61], [1, 62, 63], [1, 64, 65], [1, 66, 67], [1, 68, 69], [1, 70, 71]}

TR = 46 Prime Paths

All-DU-Paths Coverage (ADUPC):

3.1 [expr][toString()]

def(25) = def(28) = def(32) = def(33)
use(28) = use(32) = use(33) = use(34)

du pair sets:

du(25, 28, X)
du(25, 32, X)
du(25, 33, X)
du(25, 34, X)
du(28, 32, X)
du(28, 33, X)
du(28, 34, X)
du(32, 34, X)
du(33, 34, X)

TR = { [25, 26, 27, 28], [25, 26, 30, 31, 32], [25, 26, 30, 31, 33], [25, 26, 30, 31, 32, 34], [25, 26, 30, 31, 33, 34], [28, 29, 26, 30 , 31, 32], [28, 29, 26, 30, 31, 33], [28, 29, 26, 30, 31, 32, 34], [28, 29, 26, 30, 31, 33, 34], [32, 34], [33, 34]}

3.3 [Stmt][toString()]

def(16) = def(18) = def(22) = def(23)= {list }
use(18) = use(22) = use(23) = use(24) = { list }

du pair sets:

- du(16, 18, X)
- du(16, 22, X)
- du(16, 23, X)
- du(16, 24, X)
- du(18, 18, X)
- du(18, 22, X)
- du(18, 23, X)
- du(18, 24, X)
- du(22, 24, X)
- du(23, 24, X)

TR = { [16, 17, 18], [16, 20, 21, 23], [16, 20, 21, 22], [16, 20, 21, 23, 24], [16, 20, 21, 22, 24], [18, 19, 16, 17, 18], [18, 19, 16, 20, 21, 23], [18, 19, 16, 20, 21, 22], [18, 19, 16, 20, 21, 22, 24], [18, 19, 16, 20, 21, 23, 24], [22 ,24], [23,24] }

4.1. [expr] [toString()] – All Du-Path

TR ID	Test Requirement
1	[25, 26, 27, 28]
2	[25, 26, 30, 31, 32]
3	[25, 26, 30, 31, 33]
4	[25, 26, 30, 31, 32, 34]
5	[25, 26, 30, 31, 33, 34]
6	[28, 29, 26, 30 , 31, 32]
7	[28, 29, 26, 30, 31, 33]
8	[28, 29, 26, 30, 31, 32, 34]
9	[28, 29, 26, 30, 31, 33, 34]
10	[32, 34]
11	[33, 34]

4.2. [Stmt] [toString()] - Prime Paths

TR ID	Test Requirement
1	[1, 2, 3, 4, 5, 7, 8, 9, 10, 11, 14]
2	[1, 2, 3, 4, 5, 7, 8, 9, 10, 12, 14]
3	[1, 2, 3, 4, 5, 7, 8, 13, 14]
4	[1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 14]
5	[1, 2, 3, 4, 6, 7, 8, 9, 10, 12, 14]

6	[1, 2, 3, 4, 6, 7, 8, 13, 14]
7	[1, 15, 16, 17, 18, 19]
8	[1, 15, 16, 20, 21, 22, 24]
9	[1, 15, 16, 20, 21, 23, 24]
10	[16, 17, 18, 19, 16]
11	[17, 18, 19 ,16, 17, 18, 19]
12	[17, 18, 19 ,16, 20, 21, 22, 24]
13	[17, 18, 19 ,16, 20, 21, 23, 24]
14	[1, 25, 26, 27, 28]
15	[1, 25, 26, 29, 30, 31, 34]
16	[1, 25, 26, 29, 30, 31, 32, 33]
17	[26, 27, 28, 26]
18	[27, 28, 26, 27, 28]
19	[27, 28, 26, 29, 30, 31, 34]
20	[27, 28, 26, 29, 30, 31, 32, 33]
21	[31, 32, 33, 31]
22	[32, 33, 31, 32, 33]
23	[32, 33, 31, 34]
24	[1, 35, 36, 37, 39]
25	[1, 35, 36, 38, 39]
26	[1 ,40, 41]
27	[43, 44, 45, 43]
28	[44, 45, 43, 44, 45]
29	[44, 45, 43, 46, 47]
30	[1, 42, 43, 46, 47]
31	[1, 42, 43, 44, 45]
32	[50, 51, 52, 50]
33	[51, 52, 50, 51, 52]
34	[51, 52, 50, 53, 54]
35	[1, 48, 49, 50, 53, 54]
36	[1, 48, 49, 50, 51, 52]
37	[1, 55, 56, 57, 60, 61]
38	[1, 55, 56, 57, 58, 59]
39	[57, 58, 59, 57]
40	[58, 59, 57, 58, 59]
41	[58, 59, 57, 60, 61]
42	[1, 62, 63]
43	[1, 64, 65]
44	[1, 66, 67]
45	[1, 68, 69]
46	[1, 70, 71]

4.3. [BaseToken][GetType (String t)] - All Du-Path

TR ID	Test Requirement
1	[1, 2, 3]
2	[1, 2, 4]
3	[1, 2, 4, 5, 6]
4	[1, 2, 4, 5, 7]
5	[1, 2, 4, 5, 7, 8, 9]
6	[1, 2, 4, 5, 7, 8, 10]
7	[1, 2, 4, 5, 7, 8, 10, 11, 12]
8	[1, 2, 4, 5, 7, 8, 10, 11, 13]
9	[1, 2, 4, 5, 7, 8, 10, 11, 13, 14, 15]
10	[1, 2, 4, 5, 7, 8, 10, 11, 13, 14, 16]
11	[1, 2, 4, 5, 7, 8, 10, 11, 13, 14, 16, 17, 18]
12	[1, 2, 4, 5, 7, 8, 10, 11, 13, 14, 16, 17, 19]

4.4. [SymTable][getVar(String varName)] – Prime Path

TR ID	Test Requirement
1	[1, 2, 3, 4, 5, 6]
2	[1, 2, 3, 4, 5, 7, 8]
3	[1, 2, 3, 9, 10, 11]
4	[3, 4, 5, 6, 3]
5	[4, 5, 6, 3, 4, 5, 6]
6	[4, 5, 6, 3, 9, 10, 11]

4.4.1. [expr] [toString()]

Test Paths:

T1 = [1, 25, 26, 27, 28, 29, 26, 27, 28, 29, 26, 27, 28, 29, 26, 30, 31, 32, 34]
T2 = [1, 25, 26, 30, 31, 33, 34]
T3 = [1, 25, 26, 30, 31, 32, 34]
T4 = [1, 25, 26, 27, 28, 29, 26, 27, 28, 29, 26, 27, 28, 29, 26, 30, 31, 33, 34]

Test Path ID	Test Path	Covered TR	Feasible?
1	T1	[25, 26, 27, 28]	Y, expr != null
2	T3	[25, 26, 30, 31, 32]	Y, expr == null
3	T2	[25, 26, 30, 31, 33]	Y, expr == null
4	T3	[25, 26, 30, 31, 32, 34]	Y, expr == null
5	T2	[25, 26, 30, 31, 33, 34]	Y, expr == null
6	T1	[28, 29, 26, 30, 31, 32]	Y, expr != null and expr.next== null
7	T4	[28, 29, 26, 30, 31, 33]	Y, expr != null and expr.next== null
8	T1	[28, 29, 26, 30, 31, 32, 34]	Y, expr != null and expr.next== null
9	T4	[28, 29, 26, 30, 31, 33, 34]	Y, expr != null and expr.next== null
10	T3	[32, 34]	Y, expr == null
11	T2	[33, 34]	Y, expr == null

4.4.2. [Stmt] [toString()]

Test Paths:

T1 = [1, 2, 3, 4, 5, 7, 8, 9, 10, 11, 14]
T2 = [1, 2, 3, 4, 5, 7, 8, 9, 10, 12, 14]
T3 = [1, 2, 3, 4, 5, 7, 8, 13, 14]
T4 = [1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 14]
T5 = [1, 2, 3, 4, 6, 7, 8, 9, 10, 12, 14]
T6 = [1, 2, 3, 4, 6, 7, 8, 13, 14]
T7 = [1, 15, 16, 17, 18, 19, 16, 17, 18, 19, 16, 20, 21, 22, 24]
T8 = [1, 15, 16, 17, 18, 19, 16, 17, 18, 19, 16, 20, 21, 23, 24]
T9 = [1, 15, 16, 17, 20, 21, 22, 24]
T10 = [1, 15, 16, 17, 20, 21, 23, 24]
T11 = [1, 15, 16, 17, 18, 19, 16, 17, 18, 19, 16, 20, 21, 22, 24]
T12 = [1, 15, 16, 17, 18, 19, 16, 17, 18, 19, 16, 20, 21, 23, 24]
T13 = [1, 25, 26, 27, 28, 26, 27, 28, 26, 29, 30, 31, 32, 33, 31, 32, 33, 31, 34]
T14 = [1, 25, 26, 29, 30, 31, 34]
T15 = [1, 25, 26, 29, 30, 31, 32, 33]
T16 = [1, 25, 26, 27, 28, 26, 27, 28, 26, 29, 30, 31, 34]
T17 = [1, 35, 36, 37, 39]
T18 = [1, 35, 36, 38, 39]
T19 = [1 ,40, 41]
T20 = [1, 42, 43, 44, 45, 43, 44, 45, 43, 46, 47]
T21 = [1, 42, 43, 46, 47]
T22 = [1, 48, 49, 50, 51, 52, 50, 51, 52, 53, 54]
T23 = [1, 48, 49, 50, 53, 54]
T24 = [1, 55, 56, 57, 58, 59, 57, 58, 59, 57, 60, 61]
T25 = [1, 55, 56, 57, 60, 61]
T26 = [1, 62, 63]
T27 = [1, 64, 65]
T28 = [1, 66, 67]
T29 = [1, 68, 69]
T30 = [1, 70, 71]

Test Path ID	Test Path	Covered TR	Feasible?
1	T1	[1, 2, 3, 4, 5, 7, 8, 9, 10, 11, 14]	Y, case 0, typeNumber == 2 And elsestmt != null Elsestmt.typeNumber == 2
2	T2	[1, 2, 3, 4, 5, 7, 8, 9, 10, 12, 14]	Y, case 0, typeNumber == 2 And elsestmt != null Elsestmt.typeNumber != 2
3	T3	[1, 2, 3, 4, 5, 7, 8, 13, 14]	Y, case 0, typeNumber == 2 And elsestmt == null

4	T4	[1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 14]	Y, case 0, typeNumber == 2 And elsestmt != null And elsestmt.typeNumber == 2
5	T5	[1, 2, 3, 4, 6, 7, 8, 9, 10, 12, 14]	Y, case 0, typeNumber == 2 And elsestmt != null And elsestmt.typeNumber != 2
6	T6	[1, 2, 3, 4, 6, 7, 8, 13, 14]	Y, case 0, typeNumber == 2 And elsestmt == null
7	T7	[1, 15, 16, 17, 18, 19]	Y, case 1, expr != null
8	T9	[1, 15, 16, 20, 21, 22, 24]	Y, case 1, expr == null, list.length>0
9	T10	[1, 15, 16, 20, 21, 23, 24]	Y, case, expr == null, list.length <= 0
10	T11	[16, 17, 18, 19, 16]	Y, case 1, expr != null
11	T11	[17, 18, 19 ,16, 17, 18, 19]	Y, case 1, expr != null
12	T11	[17, 18, 19 ,16, 20, 21, 22, 24]	Y, case 1, expr != null and expr.next == null And list.length>0
13	T12	[17, 18, 19 ,16, 20, 21, 23, 24]	Y, case 1, expr != null and expr.next == null And list.length<=0
14	T13	[1, 25, 26, 27, 28]	Y, case 2, Fielddecl == null
15	T14	[1, 25, 26, 29, 30, 31, 34]	Y, case 2, Fielddecl == null and smtp == null
16	T15	[1, 25, 26, 29, 30, 31, 32, 33]	Y, case 2, Fielddecl ==null and smtp != null
17	T13	[26, 27, 28, 26]	Y, case 2, Fielddecl != null
18	T13	[27, 28, 26, 27, 28]	Y, case 2, Fielddecl != null
19	T16	[27, 28, 26, 29, 30, 31, 34]	Y, case 2, Fielddecl != null and Fielddecl.next == null and smtp == null
20	T13	[27, 28, 26, 29, 30, 31, 32, 33]	Y, case 2, Fielddecl != null and Fielddecl.next == null and smtp != null
21	T13	[31, 32, 33, 31]	Y, case 2, Fielddecl == null and and smtp != null
22	T13	[32, 33, 31, 32, 33]	Y, case 2, Fielddecl == null and and smtp != null
23	T13	[32, 33, 31, 34]	Y, case 2, Fielddecl == null and and smtp != null and smtp.next == null
24	T17	[1, 35, 36, 37, 39]	Y, case 3, stmt.typeNumber == 2
25	T18	[1, 35, 36, 38, 39]	Y, case 3, stmt.typeNumber != 2
26	T19	[1 ,40, 41]	Y, case 4
27	T20	[43, 44, 45, 43]	Y, case 5, name != null
28	T20	[44, 45, 43, 44, 45]	Y, case 5, name != null
29	T20	[44, 45, 43, 46, 47]	Y, case 5, name != null, name.next == null
30	T21	[1, 42, 43, 46, 47]	Y, case 5, name == null
31	T20	[1, 42, 43, 44, 45]	Y, case 5, name != null
32	T22	[50, 51, 52, 50]	Y, case 6, expr != null

33	T22	[51, 52, 50, 51, 52]	Y, case 6, expr != null
34	T22	[51, 52, 50, 53, 54]	Y, case 6, expr != null and expr.next == null
35	T23	[1, 48, 49, 50, 53, 54]	Y, case 6, expr == null
36	T22	[1, 48, 49, 50, 51, 52]	Y, case 6, expr != null
37	T25	[1, 55, 56, 57, 60, 61]	Y, case 7, expr == null
38	T24	[1, 55, 56, 57, 58, 59]	Y, case 7, expr != null
39	T24	[57, 58, 59, 57]	Y, case 7, expr != null
40	T24	[58, 59, 57, 58, 59]	Y, case 7, expr != null
41	T24	[58, 59, 57, 60, 61]	Y, case 7, expr != null and expr.next == null
42	T26	[1, 62, 63]	Y, case 8
43	T27	[1, 64, 65]	Y, case 9
44	T28	[1, 66, 67]	Y, case 10
45	T29	[1, 68, 69]	Y, case 11
46	T30	[1, 70, 71]	Y, default case

4.4.3. [BaseToken][GetType (String t)]

Test Paths:

T1 = [1, 2, 3, 20]
T2 = [1, 2, 4, 5, 6, 20]
T3 = [1, 2, 4, 5, 7, 8, 9, 20]
T4 = [1, 2, 4, 5, 7, 8, 10, 11, 12, 20]
T5 = [1, 2, 4, 5, 7, 8, 10, 11, 13, 14, 15, 20]
T6 = [1, 2, 4, 5, 7, 8, 10, 11, 13, 14, 16, 17, 18, 20]
T7 = [1, 2, 4, 5, 7, 8, 10, 11, 13, 14, 16, 17, 19, 20]

Test Path ID	Test Path	Covered TR	Feasible?
1	T1	[1, 2, 3]	Y, t == int
2	T2	[1, 2, 4]	Y, t != int
3	T2	[1, 2, 4, 5, 6]	Y, t == bool
4	T3	[1, 2, 4, 5, 7]	Y, t != bool
5	T3	[1, 2, 4, 5, 7, 8, 9]	Y, t == char
6	T4	[1, 2, 4, 5, 7, 8, 10]	Y, t != char
7	T4	[1, 2, 4, 5, 7, 8, 10, 11, 12]	Y, t == float
8	T5	[1, 2, 4, 5, 7, 8, 10, 11, 13]	Y, t != float
9	T5	[1, 2, 4, 5, 7, 8, 10, 11, 13, 14, 15]	Y, t == string
10	T6	[1, 2, 4, 5, 7, 8, 10, 11, 13, 14, 16]	Y, t != string
11	T6	[1, 2, 4, 5, 7, 8, 10, 11, 13, 14, 16, 17, 18]	Y, t == void
12	T7	[1, 2, 4, 5, 7, 8, 10, 11, 13, 14, 16, 17, 19]	Y, t != void

4.4.4. [SymTable][getVar(String varName)]

Test Paths:

T1 = [1, 2, 3, 4, 5, 6, 3, 4, 5, 6, 3, 4, 5, 7, 8]
T2 = [1, 2, 3, 9, 10, 11]
T3 = [1, 2, 3, 4, 5, 7, 8]
T4 = [1, 2, 3, 4, 5, 6, 3, 9, 10 ,11]

Test Path ID	Test Path	Covered TR	Feasible?
1	T1	[1, 2, 3, 4, 5, 6]	Y, scope != null
2	T3	[1, 2, 3, 4, 5, 7, 8]	Y, scope != null
3	T2	[1, 2, 3, 9, 10, 11]	Y, scope == null
4	T1	[3, 4, 5, 6, 3]	Y, scope != null
5	T1	[4, 5, 6, 3, 4, 5, 6]	Y, scope != null
6	T4	[4, 5, 6, 3, 9, 10, 11]	Y, scope != null and scope == null

5. Test Results with Traceability

5.1. [expr] [toString()]

Test ID	Targeted TR	MUT	Observed Output	Result
1	[25, 26, 27, 28]	toString()	(2)1	Pass
2	[25, 26, 30, 31, 32]	toString()	(2)0	Pass
3	[25, 26, 30, 31, 33]	toString()	(0)2	Pass
4	[25, 26, 30, 31, 32, 34]	toString()	(2)0	Pass
5	[25, 26, 30, 31, 33, 34]	toString()	(0)2	Pass
6	[28, 29, 26, 30, 31, 32]	toString()	(2)2	Pass
7	[28, 29, 26, 30, 31, 33]	toString()	(0)3	Pass
8	[28, 29, 26, 30, 31, 32, 34]	toString()	(2)2	Pass
9	[28, 29, 26, 30, 31, 33, 34]	toString()	(0)3	Pass
10	[32, 34]	toString()	(2)2	Pass
11	[33, 34]	toString()	(0)3	Pass

5.2. [Stmt] [toString()]

Test ID	Targeted TR	MUT	Observed Output	Result

1	[1, 2, 3, 4, 5, 7, 8, 9, 10, 11, 14]	toString()	if (1)
			{
			}
			else
			{
			}
				Pass

2	[1, 2, 3, 4, 5, 7, 8, 9, 10, 12, 14]	toString()	if (1)
			{
			}
			else
			{
			1 = 1;
			}
				Pass

3	[1, 2, 3, 4, 5, 7, 8, 13, 14]	toString()	if (1)
			{
			}
				Pass

4	[1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 14]	toString()	if (1)
			{
			while (1)
			return;
			
			}
			else
			{
			}
				Pass

5	[1, 2, 3, 4, 6, 7, 8, 9, 10, 12, 14]	toString()	if (1)
			{
			while (1)
			return;
			
			}
			else
			{
			while (1)
			return;
			
			}
				Pass

6	[1, 2, 3, 4, 6, 7, 8, 13, 14]	toString()	if (1)
			{
			while (1)
			return;
				Pass

			}	
7	[1, 15, 16, 17, 18, 19]	toString()	Hi(1, 2, 3);	Pass
8	[1, 15, 16, 20, 21, 22, 24]	toString()	foo(ab);	Pass
9	[1, 15, 16, 20, 21, 23, 24]	toString()	Hi(foo[1]);	Pass
10	[16, 17, 18, 19, 16]	toString()	Hi(1, 2, 3);	Pass
11	[17, 18, 19 ,16, 17, 18, 19]	toString()	Hi(1, 2, 3);	Pass
12	[17, 18, 19 ,16, 20, 21, 22, 24]	toString()	Hi(foo, foo1, foo2);	Pass
13	[17, 18, 19 ,16, 20, 21, 23, 24]	toString()	Hi(foo, (ct)1);	Pass

14	[1, 25, 26, 27, 28]	toString()	{
			int 1[1];
			}
				Pass

15	[1, 25, 26, 29, 30, 31, 34]	toString()	{ }	Pass

16	[1, 25, 26, 29, 30, 31, 32, 33]	toString()	{
			int 1[1];
			foo();
			}
				Pass

17	[26, 27, 28, 26]	toString()	{
			int 1[1];
			}
				Pass

18	[27, 28, 26, 27, 28]	toString()	{
			int 1[1];
			String 11[1];
			}
				Pass

19	[27, 28, 26, 29, 30, 31, 34]	toString()	{
			int 1[1];
			String 11[1];
			}
				Pass

20	[27, 28, 26, 29, 30, 31, 32, 33]	toString()	{
			int 1[1];
			String 11[1];
			foo();
			boo();
			}
				Pass

21	[31, 32, 33, 31]	toString()	{
			int 1[1];
			String 11[1];
			foo();
			boo();
			}
				Pass

22	[32, 33, 31, 32, 33]	toString()	{
			int 1[1];
			String 11[1];
			foo();
			boo();
				Pass

			}	

23	[32, 33, 31, 34]	toString()	{
			int 1[1];
			String 11[1];
			foo();
			boo();
			}
				Pass

24	[1, 35, 36, 37, 39]	toString()	while (23)
			{
			}
				Pass

25	[1, 35, 36, 38, 39]	toString()	while (23) Boo();	Pass
26	[1 ,40, 41]	toString()	2 = 2;	Pass
27	[43, 44, 45, 43]	toString()	ReadFunc;	Pass
28	[44, 45, 43, 44, 45]	toString()	ReadFunc;	Pass
29	[44, 45, 43, 46, 47]	toString()	ReadFunc;	Pass
30	[1, 42, 43, 46, 47]	toString()	nullFunc;	Pass
31	[1, 42, 43, 44, 45]	toString()	ReadFunc;	Pass
32	[50, 51, 52, 50]	toString()	Func();	Pass
33	[51, 52, 50, 51, 52]	toString()	Func();	Pass
34	[51, 52, 50, 53, 54]	toString()	Func();	Pass
35	[1, 48, 49, 50, 53, 54]	toString()	Func();	Pass
36	[1, 48, 49, 50, 51, 52]	toString()	Func();	Pass
37	[1, 55, 56, 57, 60, 61]	toString()	Func();	Pass
38	[1, 55, 56, 57, 58, 59]	toString()	Func();	Pass
39	[57, 58, 59, 57]	toString()	Func();	Pass
40	[58, 59, 57, 58, 59]	toString()	Func();	Pass
41	[58, 59, 57, 60, 61]	toString()	Func();	Pass
42	[1, 62, 63]	toString()	Johnhi;	Pass
43	[1, 64, 65]	toString()	return 2;	Pass
44	[1, 66, 67]	toString()	ID();	Pass
45	[1, 68, 69]	toString()	return();	Pass
46	[1, 70, 71]	toString()	();	Pass

5.3. [BaseToken][GetType (String t)]

Test ID	Targeted TR	MUT	Observed Output	Result
1	[1, 2, 3]	GetType()	SymTable.Type.INT	Pass
2	[1, 2, 4]	GetType()	SymTable.Type.BOOL	Pass
3	[1, 2, 4, 5, 6]	GetType()	SymTable.Type.BOOL	Pass
4	[1, 2, 4, 5, 7]	GetType()	SymTable.Type.CHAR	Pass
5	[1, 2, 4, 5, 7, 8, 9]	GetType()	SymTable.Type.CHAR	Pass

6	[1, 2, 4, 5, 7, 8, 10]	GetType()	SymTable.Type.FLOAT	Pass
7	[1, 2, 4, 5, 7, 8, 10, 11, 12]	GetType()	SymTable.Type.FLOAT	Pass
8	[1, 2, 4, 5, 7, 8, 10, 11, 13]	GetType()	SymTable.Type.STRING	Pass
9	[1, 2, 4, 5, 7, 8, 10, 11, 13, 14, 15]	GetType()	SymTable.Type.STRING	Pass
10	[1, 2, 4, 5, 7, 8, 10, 11, 13, 14, 16]	GetType()	SymTable.Type.VOID	Pass
11	[1, 2, 4, 5, 7, 8, 10, 11, 13, 14, 16, 17, 18]	GetType()	SymTable.Type.VOID	Pass
12	[1, 2, 4, 5, 7, 8, 10, 11, 13, 14, 16, 17, 19]	GetType()	Null	Pass

5.4. [SymTable][getVar(String varName)]

Test ID	Targeted TR	MUT	Observed Output	Result
1	[1, 2, 3, 4, 5, 6]	getVar()	var.name = “method”	Pass
2	[1, 2, 3, 4, 5, 7, 8]	getVar()	var.name = “no_method”	Pass
3	[1, 2, 3, 9, 10, 11]	getVar()	java.lang.Exception: third_method not defined in the SymbolTable.	Fail
4	[3, 4, 5, 6, 3]	getVar()	var.name = “fourth”	Pass
5	[4, 5, 6, 3, 4, 5, 6]	getVar()	var.name = “fourth”	Pass
6	[4, 5, 6, 3, 9, 10, 11]	getVar()	java.lang.Exception: Sixth not defined in the SymbolTable.	Fail