```
TEST CASE test001: Unary Negation
     STUDENT:
------------------------------------------------------------------------
Class Name: test001
Superclass Name:
Fields:
Constructors:
Methods:
METHOD: 1, main, test001, public, static, void
Method Parameters:
Variable Table:
VARIABLE 1, a, local, boolean
VARIABLE 2, b, local, boolean
Method Body:
Block([
Skip
,  Expr(Assign(Variable(1), Constant(Boolean-constant(True))))
,  Expr(Assign(Variable(2), <decaf_ast.AST_Unary object at
0x000002440AC685B0>))
])
------------------------------------------------------------------------
     STUDENT ERROR:


     SHOULD BE: Check print out of AST


Press Enter to continue...
```

```
TEST CASE test002: Unary Minus
      STUDENT:
------------------------------------------------------------------------
Class Name: test002
Superclass Name:
Fields:
Constructors:
Methods:
METHOD: 1, main, test002, public, static, void
Method Parameters:
Variable Table:
VARIABLE 1, a, local, int
VARIABLE 2, b, local, int
Method Body:
Block([
Skip
,  Expr(Assign(Variable(1), Constant(Integer-constant(17))))
,  Expr(Assign(Variable(2), <decaf_ast.AST_Unary object at
0x000002B48A2285B0>))
])
------------------------------------------------------------------------
      STUDENT ERROR:


      SHOULD BE: Check print out of AST


Press Enter to continue...
```

```
TEST CASE test003: Unary Plus
      STUDENT:
-----------------------------------------------------------------------
Class Name: test003
Superclass Name:
Fields:
Constructors:
Methods:
METHOD: 1, main, test003, public, static, void
Method Parameters:
Variable Table:
VARIABLE 1, a, local, int
VARIABLE 2, b, local, int
Method Body:
Block([
Skip
,  Expr(Assign(Variable(1), Constant(Integer-constant(19))))
,  Expr(Assign(Variable(2), Variable(1)))
])
-----------------------------------------------------------------------

      STUDENT ERROR:

      SHOULD BE: Check print out of AST

Press Enter to continue...
```

TEST CASE test004: Binary GTE

STUDENT:

Class Name: test004

Superclass Name:

Fields:

Constructors:

Methods:

METHOD: 1, main, test004, public, static, void

Method Parameters:

Variable Table:

VARIABLE 1, a, local, boolean

Method Body:

Block([

Skip

, Expr(Assign(Variable(1), Binary(geq, Constant(Integer-constant(17)), Constant(Integer-constant(14)))))

])

STUDENT ERROR:

SHOULD BE: Check print out of AST

Press Enter to continue...

```
TEST CASE test005: Binary GT
     STUDENT:
------------------------------------------------------------------------
Class Name: test005
Superclass Name:
Fields:
Constructors:
Methods:
METHOD: 1, main, test005, public, static, void
Method Parameters:
Variable Table:
VARIABLE 1, a, local, boolean
Method Body:
Block([
Skip
,  Expr(Assign(Variable(1), Binary(gt, Constant(Integer-constant(17)),
Constant(Integer-constant(14)))))
])
------------------------------------------------------------------------

     STUDENT ERROR:

     SHOULD BE: Check print out of AST

Press Enter to continue...
```

TEST CASE test006: Binary LTE

STUDENT:

Class Name: test006

Superclass Name:

Fields:

Constructors:

Methods:

METHOD: 1, main, test006, public, static, void

Method Parameters:

Variable Table:

VARIABLE 1, a, local, boolean

Method Body:

Block([

Skip

, Expr(Assign(Variable(1), Binary(leq, Constant(Integer-constant(17)), Constant(Integer-constant(14)))))

])

STUDENT ERROR:

SHOULD BE: Check print out of AST

Press Enter to continue...

TEST CASE test007: Binary LT

STUDENT:

Class Name: test007

Superclass Name:

Fields:

Constructors:

Methods:

METHOD: 1, main, test007, public, static, void

Method Parameters:

Variable Table:

VARIABLE 1, a, local, boolean

Method Body:

Block([

Skip

, Expr(Assign(Variable(1), Binary(lt, Constant(Integer-constant(17)), Constant(Integer-constant(14)))))

])

STUDENT ERROR:

SHOULD BE: Check print out of AST

Press Enter to continue...

```
TEST CASE test008: Binary NEQ
     STUDENT:
-------------------------------------------------------------------------
Class Name: test008
Superclass Name:
Fields:
Constructors:
Methods:
METHOD: 1, main, test008, public, static, void
Method Parameters:
Variable Table:
VARIABLE 1, a, local, boolean
Method Body:
Block([
Skip
,  Expr(Assign(Variable(1), Binary(neq, Constant(Integer-
constant(17)), Constant(Integer-constant(14)))))
])
-------------------------------------------------------------------------

     STUDENT ERROR:

     SHOULD BE: Check print out of AST

Press Enter to continue...
```

```
TEST CASE test009: Binary EQ
     STUDENT:
-------------------------------------------------------------------------
Class Name: test009
Superclass Name:
Fields:
Constructors:
Methods:
METHOD: 1, main, test009, public, static, void
Method Parameters:
Variable Table:
VARIABLE 1, a, local, boolean
Method Body:
Block([
Skip
,  Expr(Assign(Variable(1), Binary(eq, Constant(Integer-constant(17)),
Constant(Integer-constant(14)))))
])
-------------------------------------------------------------------------

     STUDENT ERROR:

     SHOULD BE: Check print out of AST

Press Enter to continue...
```

TEST CASE test010: Binary OR

STUDENT:

Class Name: test010

Superclass Name:

Fields:

Constructors:

Methods:

METHOD: 1, main, test010, public, static, void

Method Parameters:

Variable Table:

VARIABLE 1, a, local, boolean

Method Body:

Block([

Skip

, Expr(Assign(Variable(1), Binary(or, Constant(Boolean-constant(True)), Constant(Boolean-constant(False)))))

])

STUDENT ERROR:

SHOULD BE: Check print out of AST

Press Enter to continue...

TEST CASE test011: Binary AND

STUDENT:

Class Name: test011

Superclass Name:

Fields:

Constructors:

Methods:

METHOD: 1, main, test011, public, static, void

Method Parameters:

Variable Table:

VARIABLE 1, a, local, boolean

Method Body:

Block([

Skip

, Expr(Assign(Variable(1), Binary(and, Constant(Boolean-constant(True)), Constant(Boolean-constant(False)))))

])

STUDENT ERROR:

SHOULD BE: Check print out of AST

Press Enter to continue...

```
TEST CASE test012: Binary DIVISION
     STUDENT:
-------------------------------------------------------------------------
Class Name: test012
Superclass Name:
Fields:
Constructors:
Methods:
METHOD: 1, main, test012, public, static, void
Method Parameters:
Variable Table:
VARIABLE 1, a, local, int
Method Body:
Block([
Skip
,  Expr(Assign(Variable(1), Binary(div, Constant(Integer-
constant(18)), Constant(Integer-constant(6)))))
])
-------------------------------------------------------------------------

     STUDENT ERROR:

     SHOULD BE: Check print out of AST

Press Enter to continue...
```

```
TEST CASE test013: Binary MULTIPLICATION
     STUDENT:
--------------------------------------------------------------------------
Class Name: test013
Superclass Name:
Fields:
Constructors:
Methods:
METHOD: 1, main, test013, public, static, void
Method Parameters:
Variable Table:
VARIABLE 1, a, local, int
Method Body:
Block([
Skip
,  Expr(Assign(Variable(1), Binary(mult, Constant(Integer-
constant(18)), Constant(Integer-constant(6)))))
])
--------------------------------------------------------------------------

     STUDENT ERROR:


     SHOULD BE: Check print out of AST


Press Enter to continue...
```

TEST CASE test014: Binary SUBTRACTION

STUDENT:

Class Name: test014

Superclass Name:

Fields:

Constructors:

Methods:

METHOD: 1, main, test014, public, static, void

Method Parameters:

Variable Table:

VARIABLE 1, a, local, int

Method Body:

Block([

Skip

, Expr(Assign(Variable(1), Binary(sub, Constant(Integer-constant(18)), Constant(Integer-constant(6)))))

])

STUDENT ERROR:

SHOULD BE: Check print out of AST

Press Enter to continue...

```
TEST CASE test015: Binary ADDITION
     STUDENT:
-------------------------------------------------------------------------
Class Name: test015
Superclass Name:
Fields:
Constructors:
Methods:
METHOD: 1, main, test015, public, static, void
Method Parameters:
Variable Table:
VARIABLE 1, a, local, int
Method Body:
Block([
Skip
,  Expr(Assign(Variable(1), Binary(add, Constant(Integer-
constant(18)), Constant(Integer-constant(6)))))
])
-------------------------------------------------------------------------

     STUDENT ERROR:

     SHOULD BE: Check print out of AST

Press Enter to continue...
```

```
TEST CASE test016: INTEGER CONST Declaration and Assignment
      STUDENT:
------------------------------------------------------------------------
Class Name: test016
Superclass Name:
Fields:
Constructors:
Methods:
METHOD: 1, main, test016, public, static, void
Method Parameters:
Variable Table:
VARIABLE 1, a, local, int
Method Body:
Block([
Skip
,  Expr(Assign(Variable(1), Constant(Integer-constant(18))))
])
------------------------------------------------------------------------

      STUDENT ERROR:

      SHOULD BE: Check print out of AST

Press Enter to continue...
```

```
TEST CASE test017: FLOAT CONST Declaration and Assignment
      STUDENT:
-----------------------------------------------------------------------
Class Name: test017
Superclass Name:
Fields:
Constructors:
Methods:
METHOD: 1, main, test017, public, static, void
Method Parameters:
Variable Table:
VARIABLE 1, a, local, float
Method Body:
Block([
Skip
,  Expr(Assign(Variable(1), Constant(Float-constant(17.4))))
])
-----------------------------------------------------------------------

      STUDENT ERROR:


      SHOULD BE: Check print out of AST


Press Enter to continue...
```

```
TEST CASE test018: BOOLEAN TRUE CONST Declaration and Assignment
      STUDENT:
------------------------------------------------------------------------
Class Name: test018
Superclass Name:
Fields:
Constructors:
Methods:
METHOD: 1, main, test018, public, static, void
Method Parameters:
Variable Table:
VARIABLE 1, a, local, boolean
Method Body:
Block([
Skip
,  Expr(Assign(Variable(1), Constant(Boolean-constant(True))))
])
------------------------------------------------------------------------

      STUDENT ERROR:


      SHOULD BE: Check print out of AST


Press Enter to continue...
```

```
TEST CASE test019: BOOLEAN FALSE CONST Declaration and Assignment
      STUDENT:
------------------------------------------------------------------------
Class Name: test019
Superclass Name:
Fields:
Constructors:
Methods:
METHOD: 1, main, test019, public, static, void
Method Parameters:
Variable Table:
VARIABLE 1, a, local, boolean
Method Body:
Block([
Skip
,  Expr(Assign(Variable(1), Constant(Boolean-constant(False))))
])
------------------------------------------------------------------------

      STUDENT ERROR:


      SHOULD BE: Check print out of AST


Press Enter to continue...
```

```
TEST CASE test020: STRING CONST Declaration and Assignment
      STUDENT:
------------------------------------------------------------------------
Class Name: test020
Superclass Name:
Fields:
Constructors:
Methods:
METHOD: 1, main, test020, public, static, void
Method Parameters:
Variable Table:
VARIABLE 1, a, local, user(string)
Method Body:
Block([
Skip
,  Expr(Assign(Variable(1), Constant(String-constant(Hello))))
])
------------------------------------------------------------------------

      STUDENT ERROR:

      SHOULD BE: Check print out of AST

Press Enter to continue...
```

```
TEST CASE test021: AUTO-INCREMENT PRE
      STUDENT:
-----------------------------------------------------------------------
Class Name: test021
Superclass Name:
Fields:
Constructors:
Methods:
METHOD: 1, main, test021, public, static, void
Method Parameters:
Variable Table:
VARIABLE 1, a, local, int
Method Body:
Block([
Skip
,  Expr(Assign(Variable(1), Constant(Integer-constant(10))))
,  Expr(Auto(Variable(1), inc, pre))
])
-----------------------------------------------------------------------

      STUDENT ERROR:
      SHOULD BE: Check print out of AST

Press Enter to continue...
```

```
TEST CASE test022: AUTO-DECREMENT PRE
     STUDENT:
-----------------------------------------------------------------------
Class Name: test022
Superclass Name:
Fields:
Constructors:
Methods:
METHOD: 1, main, test022, public, static, void
Method Parameters:
Variable Table:
VARIABLE 1, a, local, int
Method Body:
Block([
Skip
,  Expr(Assign(Variable(1), Constant(Integer-constant(10))))
,  Expr(Auto(Variable(1), dec, pre))
])
-----------------------------------------------------------------------

     STUDENT ERROR:

     SHOULD BE: Check print out of AST

Press Enter to continue...
```

```
TEST CASE test023: AUTO-INCREMENT POST
     STUDENT:
------------------------------------------------------------------------
Class Name: test023
Superclass Name:
Fields:
Constructors:
Methods:
METHOD: 1, main, test023, public, static, void
Method Parameters:
Variable Table:
VARIABLE 1, a, local, int
Method Body:
Block([
Skip
,  Expr(Assign(Variable(1), Constant(Integer-constant(10))))
,  Expr(Auto(Variable(1), inc, post))
])
------------------------------------------------------------------------

     STUDENT ERROR:

     SHOULD BE: Check print out of AST

Press Enter to continue...
```

```
TEST CASE test024: AUTO-DECREMENT POST
     STUDENT:
------------------------------------------------------------------------
Class Name: test024
Superclass Name:
Fields:
Constructors:
Methods:
METHOD: 1, main, test024, public, static, void
Method Parameters:
Variable Table:
VARIABLE 1, a, local, int
Method Body:
Block([
Skip
,  Expr(Assign(Variable(1), Constant(Integer-constant(10))))
,  Expr(Auto(Variable(1), dec, post))
])
------------------------------------------------------------------------

     STUDENT ERROR:

     SHOULD BE: Check print out of AST

Press Enter to continue...
```

```
TEST CASE test025: IF Statement
     STUDENT:
-------------------------------------------------------------------------
Class Name: test025
Superclass Name:
Fields:
Constructors:
Methods:
METHOD: 1, main, test025, public, static, void
Method Parameters:
Variable Table:
VARIABLE 1, a, local, int
Method Body:
Block([
Skip
,  Expr(Assign(Variable(1), Constant(Integer-constant(0))))
,  <decaf_ast.AST_If object at 0x00000241A9528730>])
-------------------------------------------------------------------------

     STUDENT ERROR:

     SHOULD BE: Check print out of AST

Press Enter to continue...
```

```
TEST CASE test026: IF ELSE Statement
      STUDENT:
------------------------------------------------------------------------
Class Name: test026
Superclass Name:
Fields:
Constructors:
Methods:
METHOD: 1, main, test026, public, static, void
Method Parameters:
Variable Table:
VARIABLE 1, a, local, int
Method Body:
Block([
Skip
,  Expr(Assign(Variable(1), Constant(Integer-constant(0))))
,  <decaf_ast.AST_If object at 0x0000026017F38A00>])
------------------------------------------------------------------------

      STUDENT ERROR:


      SHOULD BE: Check print out of AST

Press Enter to continue...
```

```
TEST CASE test027: DANGLING ELSE Statement
      STUDENT:
-----------------------------------------------------------------------
Class Name: test027
Superclass Name:
Fields:
Constructors:
Methods:
METHOD: 1, main, test027, public, static, void
Method Parameters:
Variable Table:
VARIABLE 1, a, local, int
Method Body:
Block([
Skip
,  Expr(Assign(Variable(1), Constant(Integer-constant(0))))
,  <decaf_ast.AST_If object at 0x0000023B896A87C0>])
-----------------------------------------------------------------------

      STUDENT ERROR:


      SHOULD BE: Check print out of AST

Press Enter to continue...
```

```
TEST CASE test028: FOR Statement
        STUDENT:
------------------------------------------------------------------------
Class Name: test028
Superclass Name:
Fields:
Constructors:
Methods:
METHOD: 1, main, test028, public, static, void
Method Parameters:
Variable Table:
VARIABLE 1, a, local, int
VARIABLE 2, i, local, int
Method Body:
Block([
Skip
,  Skip
,  Expr(Assign(Variable(1), Constant(Integer-constant(0))))
,  <decaf_ast.AST_For object at 0x000001A7F0D46EB0>])
------------------------------------------------------------------------

        STUDENT ERROR:

        SHOULD BE: Check print out of AST

Press Enter to continue...
```

```
TEST CASE test029: WHILE Statement
      STUDENT:
-----------------------------------------------------------------------
Class Name: test029
Superclass Name:
Fields:
Constructors:
Methods:
METHOD: 1, main, test029, public, static, void
Method Parameters:
Variable Table:
VARIABLE 1, a, local, int
Method Body:
Block([
Skip
,  Expr(Assign(Variable(1), Constant(Integer-constant(0))))
,  <decaf_ast.AST_While object at 0x000002085AC38790>])
-----------------------------------------------------------------------

      STUDENT ERROR:


      SHOULD BE: Check print out of AST

Press Enter to continue...
```

```
TEST CASE test030: STATIC METHOD Definition, RETURN Statment
     STUDENT:
------------------------------------------------------------------------
Class Name: test030
Superclass Name:
Fields:
Constructors:
Methods:
METHOD: 1, add5, test030, public, static, int
Method Parameters: 1
Variable Table:
VARIABLE 1, n, formal, int
Method Body:
Block([
Return(Binary(add, Variable(1), Constant(Integer-constant(5))))
])
METHOD: 2, main, test030, public, static, void
Method Parameters:
Variable Table:
VARIABLE 1, a, local, int
Method Body:
Block([
Skip
,  Expr(Assign(Variable(1), Constant(Integer-constant(0))))
])
------------------------------------------------------------------------
     STUDENT ERROR:
     SHOULD BE: Check print out of AST
Press Enter to continue...
```

```
TEST CASE test031: STATIC METHOD Invocation Statement
     STUDENT:
-----------------------------------------------------------------------
Class Name: test031
Superclass Name:
Fields:
Constructors:
Methods:
METHOD: 1, add5, test031, public, static, int
Method Parameters: 1
Variable Table:
VARIABLE 1, n, formal, int
Method Body:
Block([
Return(Binary(add, Variable(1), Constant(Integer-constant(5))))
])
METHOD: 2, main, test031, public, static, void
Method Parameters:
Variable Table:
VARIABLE 1, a, local, int
Method Body:
Block([
Skip
,  Expr(Assign(Variable(1), Method-call(Class-reference(test031),
add5, Variable(1))))
])
-----------------------------------------------------------------------
     STUDENT ERROR:
     SHOULD BE: Check print out of AST
Press Enter to continue...
```

```
TEST CASE test032: CLASS Definition, STATIC FIELD
     STUDENT:
------------------------------------------------------------------------
Class Name: Circle
Superclass Name:
Fields:
FIELD 1, pi, Circle, public, static, float
Constructors:
Methods:
------------------------------------------------------------------------
Class Name: test032
Superclass Name:
Fields:
Constructors:
Methods:
METHOD: 1, main, test032, public, static, void
Method Parameters:
Variable Table:
VARIABLE 1, a, local, float
Method Body:
Block([
Skip
,  Expr(Assign(Variable(1), Field-access(Class-reference(Circle),
pi)))
])
------------------------------------------------------------------------
     STUDENT ERROR:
     SHOULD BE: Check print out of AST
Press Enter to continue...
```

```
TEST CASE test033: CLASS Definition, CONSTRUCTOR Definition
     STUDENT:
---------------------------------------------------------------------
Class Name: Circle
Superclass Name:
Fields:
FIELD 1, pi, Circle, public, static, float
Constructors:
CONSTRUCTOR: 1, public
Constructor Parameters: 1, 2
Variable Table:
VARIABLE 1, c, formal, float
VARIABLE 2, r, formal, float
VARIABLE 3, center, local, float
VARIABLE 4, radius, local, float
Constructor Body:
Block([
Skip
,  Skip
,  Expr(Assign(Field-access(This, center), Variable(1)))
,  Expr(Assign(Field-access(This, radius), Variable(2)))
,  Expr(Assign(Field-access(Class-reference(Circle), pi),
Constant(Float-constant(3.14159))))
])
Methods:
---------------------------------------------------------------------
Class Name: test033
Superclass Name:
Fields:
Constructors:
Methods:
```

```
METHOD: 1, main, test033, public, static, void

Method Parameters:

Variable Table:

VARIABLE 1, a, local, float

Method Body:

Block([

Skip

,  Expr(Assign(Variable(1), Field-access(Class-reference(Circle),
pi)))

])

------------------------------------------------------------------------

     STUDENT ERROR:

     SHOULD BE: Check print out of AST

Press Enter to continue...
```

```
TEST CASE test034: CLASS Definition, CONSTRUCTOR Call
     STUDENT:
---------------------------------------------------------------------
Class Name: Circle
Superclass Name:
Fields:
FIELD 1, pi, Circle, public, static, float
Constructors:
CONSTRUCTOR: 1, public
Constructor Parameters: 1
Variable Table:
VARIABLE 1, r, formal, float
VARIABLE 2, radius, local, float
Constructor Body:
Block([
Skip
,  Expr(Assign(Field-access(This, radius), Variable(1)))
,  Expr(Assign(Field-access(Class-reference(Circle), pi),
Constant(Float-constant(3.14159))))
])
Methods:
---------------------------------------------------------------------
Class Name: test034
Superclass Name:
Fields:
Constructors:
Methods:
METHOD: 1, main, test034, public, static, void
Method Parameters:
Variable Table:
VARIABLE 1, x, local, float
```

```
VARIABLE 2, c, local, user(Circle)
Method Body:
Block([
Skip
,  Skip
,  Expr(Assign(Variable(1), Constant(Float-constant(5.0))))
,  Expr(Assign(Variable(2), New-object(Circle, Variable(1))))
])
------------------------------------------------------------------------
     STUDENT ERROR:
     SHOULD BE: Check print out of AST
Press Enter to continue...
```

TEST CASE test035: CLASS Definition, NON-STATIC METHOD Definition

STUDENT:

Class Name: Circle

Superclass Name:

Fields:

FIELD 1, pi, Circle, public, static, float

Constructors:

CONSTRUCTOR: 1, public

Constructor Parameters: 1

Variable Table:

VARIABLE 1, r, formal, float

VARIABLE 2, radius, local, float

Constructor Body:

Block([

Skip

, Expr(Assign(Field-access(This, radius), Variable(1)))

, Expr(Assign(Field-access(Class-reference(Circle), pi), Constant(Float-constant(3.14159))))

])

Methods:

METHOD: 1, area, Circle, public, instance, float

Method Parameters:

Variable Table:

VARIABLE 1, rSquared, local, float

VARIABLE 2, result, local, float

Method Body:

Block([

Skip

, Skip

```
,  Expr(Assign(Variable(1), Binary(mult, Field-access(This, r), Field-
access(This, r))))

,  Expr(Assign(Variable(2), Binary(mult, Variable(1), Field-
access(Class-reference(Circle), pi))))

,  Return(Variable(2))

])

------------------------------------------------------------------------

Class Name: test035

Superclass Name:

Fields:

Constructors:

Methods:

METHOD: 1, main, test035, public, static, void

Method Parameters:

Variable Table:

VARIABLE 1, x, local, float

VARIABLE 2, c, local, user(Circle)

Method Body:

Block([

Skip

,  Skip

,  Expr(Assign(Variable(1), Constant(Float-constant(5.0))))

,  Expr(Assign(Variable(2), New-object(Circle, Variable(1))))

])

------------------------------------------------------------------------

     STUDENT ERROR:

     SHOULD BE: Check print out of AST

Press Enter to continue...
```

TEST CASE test036: CLASS Definition, NON-STATIC METHOD Invocation

STUDENT:

Class Name: Circle

Superclass Name:

Fields:

FIELD 1, pi, Circle, public, static, float

Constructors:

CONSTRUCTOR: 1, public

Constructor Parameters: 1

Variable Table:

VARIABLE 1, r, formal, float

VARIABLE 2, radius, local, float

Constructor Body:

Block([

Skip

, Expr(Assign(Field-access(This, radius), Variable(1)))

, Expr(Assign(Field-access(Class-reference(Circle), pi), Constant(Float-constant(3.14159))))

])

Methods:

METHOD: 1, area, Circle, public, instance, float

Method Parameters:

Variable Table:

VARIABLE 1, rSquared, local, float

VARIABLE 2, result, local, float

Method Body:

Block([

Skip

, Skip

```
,  Expr(Assign(Variable(1), Binary(mult, Field-access(This, r), Field-
access(This, r))))

,  Expr(Assign(Variable(2), Binary(mult, Variable(1), Field-
access(Class-reference(Circle), pi))))

,  Return(Variable(2))

])

------------------------------------------------------------------------

Class Name: test036

Superclass Name:

Fields:

Constructors:

Methods:

METHOD: 1, main, test036, public, static, void

Method Parameters:

Variable Table:

VARIABLE 1, x, local, float

VARIABLE 2, a, local, float

VARIABLE 3, c, local, user(Circle)

Method Body:

Block([

Skip

,  Skip

,  Skip

,  Expr(Assign(Variable(1), Constant(Float-constant(5.0))))

,  Expr(Assign(Variable(3), New-object(Circle, Variable(1))))

,  Expr(Assign(Variable(2), Method-call(Variable(3), area, [])))

])

------------------------------------------------------------------------

     STUDENT ERROR:

     SHOULD BE: Check print out of AST

Press Enter to continue...
```

```
TEST CASE test037: Example in Assignment, tests EXTENDS
     STUDENT:
-----------------------------------------------------------------------
Class Name: A
Superclass Name:
Fields:
FIELD 1, x, A, private, instance, int
Constructors:
CONSTRUCTOR: 1, private
Constructor Parameters:
Variable Table:
Constructor Body:
Block([
Expr(Assign(Field-access(This, x), Constant(Integer-constant(0))))
])
Methods:
METHOD: 1, f, A, private, instance, int
Method Parameters:
Variable Table:
Method Body:
Block([
Return(Binary(add, Field-access(This, x), Constant(Integer-
constant(1))))
])
METHOD: 2, g, A, public, instance, int
Method Parameters:
Variable Table:
VARIABLE 1, i, local, int
Method Body:
Block([
Skip
```

```
,  Expr(Assign(Variable(1), Method-call(This, f, [])))
,  Expr(Auto(Variable(1), inc, post))
,  Return(Variable(1))
])
----------------------------------------------------------------------
Class Name: B
Superclass Name: A
Fields:
FIELD 2, y, B, private, instance, int
FIELD 3, s, B, public, instance, user(A)
Constructors:
CONSTRUCTOR: 2, private
Constructor Parameters:
Variable Table:
Constructor Body:
Block([
Expr(Assign(Field-access(This, y), Constant(Integer-constant(2))))
,  Expr(Assign(Field-access(This, s), New-object(A, [])))
])
Methods:
METHOD: 3, f, B, public, instance, int
Method Parameters: 1
Variable Table:
VARIABLE 1, k, formal, int
Method Body:
Block([
Return(Binary(add, Method-call(Super, f, []), Variable(1)))
])
----------------------------------------------------------------------
     STUDENT ERROR:
```

```
     SHOULD BE: Check print out of AST
Press Enter to continue...
```

```
TEST CASE test038: hello_world.decaf

     STUDENT:

     STUDENT ERROR: Traceback (most recent call last):

  File
"C:\Users\cmkan\Dropbox\StonyBrook\Teaching\008_2022_Spring\CSE304\Ass
ignments\A03\grading\cse304_a03_KZ\decaf_checker.py", line 23, in
<module>

    main()

  File
"C:\Users\cmkan\Dropbox\StonyBrook\Teaching\008_2022_Spring\CSE304\Ass
ignments\A03\grading\cse304_a03_KZ\decaf_checker.py", line 16, in main

    print(parser.parse(progtext, lexer=lexer))

  File "C:\Users\cmkan\AppData\Roaming\Python\Python39\site-
packages\ply-3.11-py3.9.egg\ply\yacc.py", line 333, in parse

    return self.parseopt_notrack(input, lexer, debug, tracking,
tokenfunc)

  File "C:\Users\cmkan\AppData\Roaming\Python\Python39\site-
packages\ply-3.11-py3.9.egg\ply\yacc.py", line 1120, in
parseopt_notrack

    p.callable(pslice)

  File
"C:\Users\cmkan\Dropbox\StonyBrook\Teaching\008_2022_Spring\CSE304\Ass
ignments\A03\grading\cse304_a03_KZ\decaf_parser.py", line 23, in
p_program

    p[0] = AST_Program(p[1])

  File
"C:\Users\cmkan\Dropbox\StonyBrook\Teaching\008_2022_Spring\CSE304\Ass
ignments\A03\grading\cse304_a03_KZ\decaf_ast.py", line 149, in
__init__

    raise InvalidVariableReferenceException

decaf_ast.InvalidVariableReferenceException

     SHOULD BE: Check print out of AST

Press Enter to continue...
```

```
TEST CASE test039: nrfib.decaf

     STUDENT:

     STUDENT ERROR: Traceback (most recent call last):

  File
"C:\Users\cmkan\Dropbox\StonyBrook\Teaching\008_2022_Spring\CSE304\Ass
ignments\A03\grading\cse304_a03_KZ\decaf_checker.py", line 23, in
<module>

    main()

  File
"C:\Users\cmkan\Dropbox\StonyBrook\Teaching\008_2022_Spring\CSE304\Ass
ignments\A03\grading\cse304_a03_KZ\decaf_checker.py", line 16, in main

    print(parser.parse(progtext, lexer=lexer))

  File "C:\Users\cmkan\AppData\Roaming\Python\Python39\site-
packages\ply-3.11-py3.9.egg\ply\yacc.py", line 333, in parse

    return self.parseopt_notrack(input, lexer, debug, tracking,
tokenfunc)

  File "C:\Users\cmkan\AppData\Roaming\Python\Python39\site-
packages\ply-3.11-py3.9.egg\ply\yacc.py", line 1120, in
parseopt_notrack

    p.callable(pslice)

  File
"C:\Users\cmkan\Dropbox\StonyBrook\Teaching\008_2022_Spring\CSE304\Ass
ignments\A03\grading\cse304_a03_KZ\decaf_parser.py", line 23, in
p_program

    p[0] = AST_Program(p[1])

  File
"C:\Users\cmkan\Dropbox\StonyBrook\Teaching\008_2022_Spring\CSE304\Ass
ignments\A03\grading\cse304_a03_KZ\decaf_ast.py", line 149, in
__init__

    raise InvalidVariableReferenceException

decaf_ast.InvalidVariableReferenceException

     SHOULD BE: Check print out of AST

Press Enter to continue...
```

```
TEST CASE test040: rfib.decaf

     STUDENT:

     STUDENT ERROR: Traceback (most recent call last):

  File
"C:\Users\cmkan\Dropbox\StonyBrook\Teaching\008_2022_Spring\CSE304\Ass
ignments\A03\grading\cse304_a03_KZ\decaf_checker.py", line 23, in
<module>

    main()

  File
"C:\Users\cmkan\Dropbox\StonyBrook\Teaching\008_2022_Spring\CSE304\Ass
ignments\A03\grading\cse304_a03_KZ\decaf_checker.py", line 16, in main

    print(parser.parse(progtext, lexer=lexer))

  File "C:\Users\cmkan\AppData\Roaming\Python\Python39\site-
packages\ply-3.11-py3.9.egg\ply\yacc.py", line 333, in parse

    return self.parseopt_notrack(input, lexer, debug, tracking,
tokenfunc)

  File "C:\Users\cmkan\AppData\Roaming\Python\Python39\site-
packages\ply-3.11-py3.9.egg\ply\yacc.py", line 1120, in
parseopt_notrack

    p.callable(pslice)

  File
"C:\Users\cmkan\Dropbox\StonyBrook\Teaching\008_2022_Spring\CSE304\Ass
ignments\A03\grading\cse304_a03_KZ\decaf_parser.py", line 23, in
p_program

    p[0] = AST_Program(p[1])

  File
"C:\Users\cmkan\Dropbox\StonyBrook\Teaching\008_2022_Spring\CSE304\Ass
ignments\A03\grading\cse304_a03_KZ\decaf_ast.py", line 149, in
__init__

    raise InvalidVariableReferenceException

decaf_ast.InvalidVariableReferenceException

     SHOULD BE: Check print out of AST

Press Enter to continue...
```

```
TEST CASE test041: IntList.decaf
     STUDENT:
---------------------------------------------------------------------
Class Name: IntList
Superclass Name:
Fields:
FIELD 1, value, IntList, private, instance, int
FIELD 2, next, IntList, private, instance, user(IntList)
Constructors:
CONSTRUCTOR: 1, public
Constructor Parameters:
Variable Table:
Constructor Body:
Block([
])
Methods:
METHOD: 1, create_list, IntList, public, static, user(IntList)
Method Parameters: 1
Variable Table:
VARIABLE 1, v, formal, int
VARIABLE 2, new_element, local, user(IntList)
Method Body:
Block([
Skip
,  Expr(Assign(Variable(2), New-object(IntList, [])))
,  Expr(Assign(Field-access(Variable(2), value), Variable(1)))
,  Expr(Assign(Field-access(Variable(2), next), Constant(Null-
constant)))
,  Return(Variable(2))
])
METHOD: 2, insert, IntList, public, instance, user(IntList)
```

```
Method Parameters: 1
Variable Table:
VARIABLE 1, v, formal, int
VARIABLE 2, new_element, local, user(IntList)
Method Body:
Block([
Skip
,  Expr(Assign(Variable(2), Method-call(Class-reference(IntList),
create_list, Variable(1))))
,  Expr(Assign(Field-access(Variable(2), next), This))
,  Return(Variable(2))
])
METHOD: 3, search, IntList, public, instance, boolean
Method Parameters: 1
Variable Table:
VARIABLE 1, v, formal, int
Method Body:
Block([
<decaf_ast.AST_If object at 0x000001E7F8EFEB50>])
METHOD: 4, length, IntList, public, instance, int
Method Parameters:
Variable Table:
Method Body:
Block([
<decaf_ast.AST_If object at 0x000001E7F8F02790>])
--------------------------------------------------------------------------
     STUDENT ERROR:
     SHOULD BE: Check print out of AST
Press Enter to continue...
```

```
TEST CASE error01: Duplicate Class Name

     STUDENT:

     STUDENT ERROR: Traceback (most recent call last):

  File
"C:\Users\cmkan\Dropbox\StonyBrook\Teaching\008_2022_Spring\CSE304\Ass
ignments\A03\grading\cse304_a03_KZ\decaf_checker.py", line 23, in
<module>

    main()

  File
"C:\Users\cmkan\Dropbox\StonyBrook\Teaching\008_2022_Spring\CSE304\Ass
ignments\A03\grading\cse304_a03_KZ\decaf_checker.py", line 16, in main

    print(parser.parse(progtext, lexer=lexer))

  File "C:\Users\cmkan\AppData\Roaming\Python\Python39\site-
packages\ply-3.11-py3.9.egg\ply\yacc.py", line 333, in parse

    return self.parseopt_notrack(input, lexer, debug, tracking,
tokenfunc)

  File "C:\Users\cmkan\AppData\Roaming\Python\Python39\site-
packages\ply-3.11-py3.9.egg\ply\yacc.py", line 1120, in
parseopt_notrack

    p.callable(pslice)

  File
"C:\Users\cmkan\Dropbox\StonyBrook\Teaching\008_2022_Spring\CSE304\Ass
ignments\A03\grading\cse304_a03_KZ\decaf_parser.py", line 23, in
p_program

    p[0] = AST_Program(p[1])

  File
"C:\Users\cmkan\Dropbox\StonyBrook\Teaching\008_2022_Spring\CSE304\Ass
ignments\A03\grading\cse304_a03_KZ\decaf_ast.py", line 89, in __init__

    raise MultipleClassDefinitionException

decaf_ast.MultipleClassDefinitionException

     SHOULD BE: Check print out of AST

Press Enter to continue...
```

```
TEST CASE error02: Duplicate Field Name

     STUDENT:

     STUDENT ERROR: Traceback (most recent call last):

  File
"C:\Users\cmkan\Dropbox\StonyBrook\Teaching\008_2022_Spring\CSE304\Ass
ignments\A03\grading\cse304_a03_KZ\decaf_checker.py", line 23, in
<module>

    main()

  File
"C:\Users\cmkan\Dropbox\StonyBrook\Teaching\008_2022_Spring\CSE304\Ass
ignments\A03\grading\cse304_a03_KZ\decaf_checker.py", line 16, in main

    print(parser.parse(progtext, lexer=lexer))

  File "C:\Users\cmkan\AppData\Roaming\Python\Python39\site-
packages\ply-3.11-py3.9.egg\ply\yacc.py", line 333, in parse

    return self.parseopt_notrack(input, lexer, debug, tracking,
tokenfunc)

  File "C:\Users\cmkan\AppData\Roaming\Python\Python39\site-
packages\ply-3.11-py3.9.egg\ply\yacc.py", line 1120, in
parseopt_notrack

    p.callable(pslice)

  File
"C:\Users\cmkan\Dropbox\StonyBrook\Teaching\008_2022_Spring\CSE304\Ass
ignments\A03\grading\cse304_a03_KZ\decaf_parser.py", line 23, in
p_program

    p[0] = AST_Program(p[1])

  File
"C:\Users\cmkan\Dropbox\StonyBrook\Teaching\008_2022_Spring\CSE304\Ass
ignments\A03\grading\cse304_a03_KZ\decaf_ast.py", line 132, in
__init__

    resolve_class(i)

  File
"C:\Users\cmkan\Dropbox\StonyBrook\Teaching\008_2022_Spring\CSE304\Ass
ignments\A03\grading\cse304_a03_KZ\decaf_ast.py", line 120, in
resolve_class

    raise MultipleFieldDefinitionException

decaf_ast.MultipleFieldDefinitionException
```

```
      SHOULD BE: Check print out of AST
Press Enter to continue...
```

```
TEST CASE error03: Duplicate Variable
      STUDENT:
Syntax error at line 5 column 21 token ')'
------------------------------------------------------------------------
------------------------------------------------------------------------
Error parsing file
      STUDENT ERROR:
      SHOULD BE: Check print out of AST
Press Enter to continue...
```