Custom Test Cases with Outputs:

Standard Output:

```
PRINT(“STANDARD OUTPUT”)
PRINT(6.4, TRUE, 2, “OK”)
```

```
STANDARD OUTPUT
6.4 TRUE 2 OK
```

Variables:

```
STRING a = "g o"
INT b = 1111
DOUBLE c = 2.54321
BOOL d = TRUE
INT e = -1
STRING f = "end"
PRINT(a)
PRINT(b,c)
PRINT(d,e)
PRINT(f)
INT e = 5
```

```
g o
1111 2.54321
TRUE -1
end
RedeclarationError
```

Expressions:

```
STRING mystring = "theory" + (“ of ” + “automata”) +
“315”
INT a = -6
INT b = 2
b++
PRINT(mystring)
DOUBLE c = (1.5-0.5+2)
DOUBLE determinant = b ^ 2 - 4 * a * c
DOUBLE quadratic_root1 = (-b + determinant^(1/2)) /
( 2.0*a )
PRINT(quadratic_root1)

BOOL d = TRUE
PRINT(NOT TRUE == (NOT (NOT d)) AND (TRUE != 0))
PRINT(4 + “A”)
PRINT(“Hi”)
```

```
theory of
automata 315
-0.5
FALSE
TypeError
```

If-elseif-else:

```
BOOL isRaining = FALSE
BOOL isSnowing = TRUE
INT temp = 45
IF (isRaining == 0)
{
      IF(temp > 45) {
            PRINT(“Wear lightweight raincoat”)
      }
      ELSEIF(temp == 45) {
            PRINT(“Wear lightweight raincoat”)

            IF (isSnowing == 1) {
                PRINT(“Passed!”)
            }
      }
      ELSE {
            PRINT(“Wear fleece and raincoat”)
      }
}
ELSE IF (isSnowing != FALSE)
{
      IF(temp > 20) {
            PRINT(“Wear soft shell jacket”)
      }
      ELSEIF (temp >= 0) {
            PRINT(“Wear down jacket”)
      }
      ELSE {
            PRINT(“Wear base layers and down jacket”)
      }
}
ELSE {
      print(“It is hard to come up with interesting
examples”)
}
```

```
Wear lightweight
raincoat
Passed!
```

Do-while Loops:

```
INT i=0
DO {
      INT j=0
      DO
      {
            INT k=0
            DO
            {
                  BOOL l = FALSE
                  DO
                  {
                        PRINT("(", i, ",", j, ",", k,
                  ",", l, ")")
                        l = TRUE
                  } WHILE (l != TRUE)
                  k++
            } WHILE (k<2)
            j++
      } WHILE (j<2)
      i++

} WHILE (i<2)
```

```
( 0 , 0 , 0 , FALSE )
( 0 , 0 , 1 , FALSE )
( 0 , 1 , 0 , FALSE )
( 0 , 1 , 1 , FALSE )
( 1 , 0 , 0 , FALSE )
( 1 , 0 , 1 , FALSE )
( 1 , 1 , 0 , FALSE )
( 1 , 1 , 1 , FALSE )
```

For Loops:

```
FOR (INT i = 0; i < 2; i++)
{
      FOR (INT j = 0; j <= 1; j = j + 1)
      {
            FOR (INT k = 0; k <= 1; k++)
            {
                  FOR (BOOL l = FALSE; l != TRUE; l
      =TRUE)
                  {
                        PRINT("(", i, ",", j, ",", k,
            ",", l, ")")
                  }

            }
      }
}
```

```
( 0 , 0 , 0 , FALSE )
( 0 , 0 , 1 , FALSE )
( 0 , 1 , 0 , FALSE )
( 0 , 1 , 1 , FALSE )
( 1 , 0 , 0 , FALSE )
( 1 , 0 , 1 , FALSE )
( 1 , 1 , 0 , FALSE )
( 1 , 1 , 1 , FALSE )
```

Lists:

```
LIST A = [0,1,2,3,4,5,6]
LIST B = [“OK”]
PRINT(B)
PRINT(A.slice(3, 6))
B.push(“Get Ready!”)
B.push(“OK!”)
B.push(“Passed!”)
B.push(“Done!”)
PRINT(B.index(3))
STRING C = “Well ” + B.pop(3)
PRINT(C)
PRINT(B)
LIST D = [7]
A.PUSH(D.POP(0))
PRINT(A)
PRINT(B[7])
```

```
[“OK”]
[3,4,5]
Passed!
Well Passed!
[“OK”, “Get Ready!”,
“OK!”, “Done!”]
[0,1,2,3,4,5,6,7]
IndexOutOfBoundsErro
r
```

Structs:

```
STRUCT BookStruct {
   STRING title
   STRING author
   BOOL is_published
   STRING subject
   INT book_id
};

BookStruct Book1
BookStruct Book2
BookStruct Book3

Book3.is_published = False

Book1.title = "The Theory of Computation"
Book1.author = "Michael Sipser"
Book1.subject = "Theory of Computation"
Book1.book_id = 1337
Book1.is_published = True

Book2.title = "Race Against the Machine"
Book2.author = "Andrew McAfee"
Book2.subject = "Digital Technology"
Book2.book_id = 92315


PRINT(Book1.title)
PRINT(Book1.author)
PRINT(Book1.subject)
PRINT(Book1.book_id)
PRINT(Book1.is_published)
PRINT(“”)
PRINT(Book2.title)
PRINT(Book2.author)
PRINT(Book2.subject)
PRINT(Book2.book_id)


Book3.title = Book1.subject + “2”
PRINT(Book3.title)

PRINT(Book2.publisher)
```

```
The Theory of
Computation
Michael Sipser
Theory of Computation
True

Race Against the Machine
Andrew McAfee
Digital Technology
92315
Theory of Computation 2
AttributeError
```

Tasks Breakdown: **(Total Marks: 20)**

Compulsory (Easy): (Marks: 6)

Variables (2):

a. Declaration, assignment, access
b. Static-typing (types restricted to: int, double, char, string, bool)
c. Initialisation and declaration of a variable with the name of a pre existing variable should generate an error.

Expressions (3):

a. Numerical Operators: (+ , - , / , * , ^, % , ++, --)
b. Logical Operators (<, >, <=,>=, !=, ==, NOT, AND, OR)
c. Nested parentheses
d. Type (e.g. for String + Int) and division by 0 errors

Standard Output (1)

a. Single object is printed with a line break.
b. Multiple objects separated by a delimiter (e.g. comma as in Python) are printed with spaces in between and a line break at the end.

Medium: (Marks: 6)

Attempt only the one allotted to you:

If-elseif-else statements:

a. Can be nested
b. Can be an if statement, or just an if-elseif-elseif-...else or if-elseif or an if-else

Do-While Loops:

a. Can be nested

For Loops:

a. Can be nested

Hard: (Marks: 8)

Attempt only the one allotted to you:

List:

a. Declaration, assignment, access
b. Access outside the list-size should also give an error (e.g. Index Out of Bounds)
c. list.pop(index of item to remove) // list.pop(0) removes the head of the list and returns it
d. list.push(value) // appends the value at the end of the list
e. list.index(index) // returns the value at that index

f. list.slice(start, end) // end index excluded e.g. [6, 3, 1, 5, 2].slice(1, 4) returns [3, 1, 5]

Structs:

a. Define a struct
b. Create an object with the defined struct type
c. Access the variables/attributes declared inside the struct and assign values
d. AttributeError when a non-existing attribute is accessed