

## *Testing Report*

Carlone & Nordby

Parser, Scanner, and Syntax Tree:

For these three parts of the project, no errors are found. The programmers are able to see the expected results when printing the syntax tree, which is a good indicator that the parser, scanner, and syntax tree were created correctly.

Semantic Checking:

COMPLETED:

### 1. Semantic rules for Scoping

- 1.1. Local objects cannot be declared more than once
- 1.2. Local objects hide non-local objects with the same name
- 1.3. Non-local objects should be visible from inner scopes (unless a local object of the same name exists)
- 1.4. Function and procedure names exist in the scope they are defined (and not in their own scopes)
- 1.5. Local objects cease to exist once their scopes cease to exist

### 2. Semantic rules for Expressions

- 2.1. Expressions return typed-values
- 2.2. Objects must be declared before they used in expressions
- 2.3. Objects of different types cannot appear in the same expression (no type promotions)

### 3. Semantic rules for Statements

3.1. Statements do not return values

3.2. The test expression for IF-THEN, IF-THEN-ELSE, WHILE-DO must be Boolean-valued;

note that the Boolean type must be implicitly defined

3.3. The ELSE clause always binds to the closest IF (resolution of the dangling ELSE problem)

3.4. The variable type in FOR-DO must match the types of lower bound and upper bound expressions

### 4. Semantic rules for Arrays

4.1. Non-integer valued expressions cannot be used for indexing arrays

ALL BELOW INCOMPLETE

### 5. Semantic rules for Functions

5.1. Function calls return values of type Integer or Real (Partly Done)

5.2. Function must contain a "return" statement within its own body;

this is of the form: <function\_name> := <expression>

5.3. Functions must accept exactly the same number of arguments as is

declared in its header, with the correct sequence of types

5.4. Functions are not allowed to update the value of nonlocal objects (via assignment statements)

## 6. Semantic rules for Procedures

6.1. Procedure calls do not return values (Partly done)

6.2. Procedures must accept exactly the same number of arguments as is declared in its header, with the correct sequence of types

### Code Generation:

Sadly, the programmers were not able to complete any test to indicate that the code generation is complete or correct, except for creating the file for the assembly to be hosted in.