Compiler Project

Developers:
Sooraj Tom, 111501036
Adrian McDonald Tariang, 111501001

In Brief:

A Compiler for a simplified version of C, which creates an Abstract Parse Tree and Translates to JavaScript.

This can be further executed using Node.js or the Web Browser.

Compiling the Compiler

- $ make
- $ ./compiler filename

Testing the output

- node filename

What we planned

To make a compiler for a simple language that comprises of:

- Arithmetic operations
- Boolean logic
- Conditional statements
- Loops
- Basic I/O
- User defined functions with arguments

What we were able to achieve:

We were able to make a language with all the functionality except the user defined functions with arguments.

Work in Progress

- Pretty Printing
- Interpreter for our language

Syntax

- **Keywords**: int string if then else end while do for continue break print readint fun call
- **Operators**: (binary) + - * / % && || := = != (unary) - !
- Individual statements are separated with ;
- **Assignment operation**, a := 1
- **Conditional Statements**:
  if <expr> then <stmnts> else <stmnts> end
  If <expr> then <stmnts> end
- **Loops:** while <expr> do <stmnts> end
  for <stmnt> : <expr> : <stmnt> do <stmnts> end
- **Reading input**: a := readint “Enter a number”
- **Writing output**: print “Hello”
- **Defining functions:**: fun <function_name> : <stmnts> end
- **Calling functions**: call <function_name>