

G++

G++ is a Gebze Technical University programming language with:

- Lisp like syntax
- Imperative, non-object oriented
- Static scope, static binding, strongly typed, ...

G++ Interpreter

- Starting coffee without an input file...

```
$ g++
```

```
> _
```

\\READ-EVAL-PRINT loop starts here...

- Starting coffee with an input file...

```
$ g++ myprogram.coffee
```

\\READ-EVAL-PRINT everything in the file...

```
> _
```

\\READ-EVAL-PRINT loop starts here...

G++ - Syntax

- Keywords: *and, or, not, equal, append, concat, set, deffun, for, while, if, exit*
- Operators: *+, -, /, *, (,), ***
- Terminals:
 - *Keywords, operators, 0-9*
 - *BinaryValue -> true | false*
 - *IntegerValue -> [-]*[1-9]*[0-9]+*
 - *Id - [a-zA-z]+*

G++ - Syntax

- Non-terminals:
 - START, INPUT, EXPLISTI, EXPI, EXPB, ...

G++ - Syntax

- START -> INPUT
- INPUT -> EXPI | EXPLISTI

G++ - Syntax

- Lists
 - LISTVALUE -> '(VALUES) | '() | null
- VALUES -> VALUES IntegerValue | IntegerValue

G++ - Syntax

- An expression returns either a binary, integer or integer list (prints the corresponding value, e.g. “true”, “123”, “(12,13,14)”)
- Expressions:
 - EXPI -> (+ EXPI EXPI) |
(- EXPI EXPI) | (* EXPI EXPI) |
(/ EXPI EXPI) | Id | IntegerValue | (Id EXPLISTI)
 - EXPB -> (and EXPB EXPB) |
(or EXPB EXPB) | (not EXPB) |
(equal EXPB EXPB) | (equal EXPI EXPI) | BinaryValue
 - EXPLISTI -> (concat EXPLISTI EXPLISTI) | (append EXPI EXPLISTI) | LISTVALUE | null

G++ - Syntax

- Assignment:
 - EXPI -> (set Id EXPI)
 - Imperative, therefore EXPI will be evaluated first...

G++ - Syntax

- Functions:
 - Definition:
 - EXPI -> (deffun Id IDLIST EXPLISTI)
 - Call:
 - EXPI -> (Id EXPLISTI)
 - Parameter passing by value
 - Returning the value of the last expression
 - *Note that function definition is an expression always returning 0*

G++ - Syntax

- Control Statements:
 - EXPI -> (if EXPB EXPLISTI)
 - EXPI -> (if EXPB EXPLISTI EXPLISTI)
 - EXPI -> (while (EXPB) EXPLISTI)
 - EXPI -> (for (Id EXPI EXPI) EXPLISTI)

G++ – Variables

- EXPI -> (defvar Id EXPI) // defining a variable
- EXPI -> (set Id EXPI) // setting a variable
 - Scope:
 - Static, lexical scope (shadowing)
 - Binding:
 - Static binding
 - Typing:
 - Strong typing...

Example Programming in G++

```
$ g++  
> (deffun sumup (x)  
  (if (equal x 0)  
    1  
    (+ x (sumup (- x 1)))  
  )  
)  
> (sumup 4)  
10  
> (exit)  
$_
```