

Reading

Read Sections 7.1–7.6 of our textbook *Compilers*.

Written Assignment 7

There is no written assignment from our textbook.

Programming Assignment 7

Through assignment 6, our target language has the syntax described below, with semantics as described in these past assignments:

<i>prog</i>	→	<i>expr</i> ⁺
<i>expr</i>	→	DOUBLE
		BOOLEAN
		ID
		(' RATOR <i>expr</i> * ')
		(' 'def' ID <i>expr</i> ')
		(' 'if' <i>expr</i> ₁ <i>expr</i> ₂ <i>expr</i> ₃ ')
		(' 'let' <i>letvardec</i> <i>expr</i> ')
		(' 'while' <i>expr</i> ₁ <i>expr</i> ₂ ')
		(' 'begin' <i>expr</i> ⁺ ')
		(' 'fun' ID <i>expr</i> ')
		(' 'call' <i>expr</i> ₁ <i>expr</i> ₂ ')
<i>letvardec</i>		[' ID <i>expr</i> ']
BOOLEAN	→	'true' 'false'
RATOR	→	ARITHMETIC RELATIONAL BOOLEAN
ARITHMETIC	→	'+' '-' '*' '/'
RELATIONAL	→	'=' '>' '<'
BOOLEAN	→	'&' '!' '!'

Programming project 7 asks you to define your own addition to this target language and to implement it! You will need to define the syntax and semantics for your new constructs, and then implement them your interpreter. Alternatively, this may involve modifying existing syntax and semantics. I urge you to work through some examples by hand to better understand what you have in mind, and to serve as test cases for your implementation. Here are some possibilities, although you're welcome to devise a new feature not on this list.

- Add new looping constructs, such as a *do-while* or a *for* loop.
- Add a *break* and a *continue* statement for enhancing iterative flow of control.
- Add a compound data type, such as lists, arrays, or tuples. Each would require support for constructing instances of the data type (including specifying instances as literals) and for accessing its elements.
- Add support for multi-argument function definition and calls, and/or for multi-variable *let* declarations.
- Add a feature that optionally supports dynamic scoping.
- Add a 'syntactic sugar' for naming and defining new functions in a single expression.
- Add constructs for raising and handling exceptions.

As an alternative to the above, you're welcome to either (a) enhance the source language we've been developing in class as part of my talks, or (b) develop a domain specific language (DSL) using Antlr and Java.

Whatever you do for this assignment, I encourage you to present your work to the class. (This is not a requirement but an invitation with encouragement.) If you do wish to present your work, let me know in advance so I can schedule you in.