Milestone 4: Constant Only Calculations - DUE 02/28/14 (11:59pm)

Objective

Objective 1 is to produce output suitable for use by *gforth*.

Objective 2 is to compile constant only operations. **No program should contain *let* statements or *variables* for this milestone.**

Objective 3 to give you experience using semantic formalisms is designing the code generator.

Objective 4 is to test the actual use of the parse tree.

Objective 5 is to gain experience in modifying and enhancing your programs.

Professional Methods and Values

Design and testing. Use of formal method.

Assignment

We are now ready to have a very minimal compiler. We have developed the parse tree and we can process that tree. We now finish the first development cycle by putting in the code generator.

Performance Objectives

1. Develop a formal definition of the code generation algorithm from the intended (naïve) semantics for IBTL based on the previously developed parser.
2. Test the resulting program for correctness based on the definition.

For this milestone, you must generate *gforth* code and run that code to show the result. Your tests must show that the operations are correctly implemented based on their customary definitions in boolean, integer, floating point and string operations. **A basic approach for testing these primitive operators is to choose *simple* values that you can calculate easily. It is not a valid test case if you can't tell the value ahead of time.** Therefore, you should keep the following in mind:

1. Keep the code short.

Milestone Report

The milestone report should concentrate on the development of the program. It must include any design decisions that you make that are not also made in class; all such decisions should be documented by explaining why the decision was made. ** Remember to include handwritten solutions for the design, specification, processing, testing, and retrospective sections in your milestone report!**

Operator Table

Primitive Data	Operation Name	Number
Type	See Equivalent *C* Definition	of Operands
Boolean		
	and	2
	or	2
	not	1
Integers		
	plus	2
	minus	2
	negate	1
	times	2
	divide	2
	remainder	2
	power	2
	less than (equal)	2
	greater than (equal)	2

	(not) equal	2
Floating		
	plus	2
	minus	2
	negate	1
	times	2
	divide	2
	remainder	2
	power	2
	sin	1
	cos	1
	tan	1
	less than (equal)	2
	greater than (equal)	2
	(not) equal	2
Strings		
	concat	2