Compiler Lab (MCA) Jan-April 2012

Grading Policy :

Continuous Evaluation (100%)

The class will be divided into groups of 3 students each (of your own choice). Evaluations will be held for each group according to the schedule below. The lab follows flexible schedule. Each group is expected to complete the stages as per schedule. However if you need more time to complete, you can proceed at your pace. You will be evaluated based on what you have submitted.

Integrity Policy: Any case of copy if found out will qualify for an automatic F grade.

The laboratory will be held in Network Systems Lab, CSED on Fridays, 2:00 PM to 5:00 PM.

Schedule:

Jan 06 : Implement a simple calculator with tokens recognized using flex and parsing and semantic actions done using Bison. Print the postfix and prefix form of the expression.

Jan 13 : Extend the simple calculator mentioned above with semantic actions to build an expression tree. Write code for performing post order and in-order traversal on the tree. You can use the expression tree structure here . You won't need all fields of the structure now, but they will be useful later.

Jan 20 : Write an evaluator for the expression tree above and add facility for integer valued variables to be declared and used. You will need to implement a symbol table. The symbol table can be a linked list of the structure given here . Again, you won't use all the entries of the structure now.

Jan 27 : Add array variables to the expression evaluator with variables above. The symbol table structure given allows you to add arrays easily.

Feb 10 : Add "if-then-else" and "while" constructs to the calculator with array variables and modify the evaluator to handle these constructs. Now you have an interpretter for a full fledged programming system with only integer type variables.

Feb 17 : Allow boolean type variables also to be declared in the programming language constructed so far. You must check for types before constructing the expression tree when types are added.

Feb 24 : Add code generation module to to get target code generated for the SIM machine for the programming language you have developed so far.

March 02: Implement parser for function declarations and definitions.

March 16: Function call implementation - code generation.

March 30: Function Call implementation - code generation (contd.)

April 13, 20: Buffer days for pending submissions

Page maintained by: K. Murali Krishnan(kmurali@nitc.ac.in)