Course Administration

- Exercise 1 Redo
- Exercise 2 Preliminary Exercise
Regular Expressions

- Alphabet – the symbols that actually appear in the lexeme
- Special symbols to define the regular expression
  - ( ) : grouping
  - * : 0 or more occurrences of a pattern
  - + : 1 or more occurrences of a pattern
  - | : indicates alternatives
  - λ : indicates nothing (lambda)
Regular Expression Examples

- Alphabet = \{a,b,c\}
- Examples
  - a (b | c) a →
  - a^+ (b | c) a^+ -->
  - a (b | c)^* a →
  - abc^*ba →
  - (a|b|c|\lambda)((ab^*c)|(cb^*a))^+ →
Regular Expression for User Defined Names
A regular expression for unsigned integer numeric literals
A regular expression for signed integer numeric literals
Deterministic Finite Automata

- States
  - 5

- Transitions
  - a

- Start state
  - 1

- Final state
  - 9
DFAs for Examples

- a \ (b \mid c) \ a
- a^+ \ (b \mid c) \ a^+
- a \ (b \mid c)^* \ a
- abc^*ba
- (a|b|c|\lambda)((ab^*c)|(cb^*a))^+
Regular Expression for Numeric Literals

- Regular expression for general class of numeric literals signed/unsigned and integer/real
- Alphabet = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, -, +, .}
- Regular Expression

- How do you recognize the end of a numeric literal?
Things to think about

- What does the g++ compiler do with
  - `int i = "Hello";` ?

- What about
  - `char c = 'H';` ?
  - `char c = 'H';` ?

- What does the g++ compiler do with
  - `int i = -000;` ?
  - `cout << i << endl;`