**CS 3160 Concepts of Programming Languages**

**Fall 2018 Assignment 2**

**Due 19-5-2018**

1.a) Given a BNF, e ::= n | e+e | e-e | e \* e | (e)   which describes the syntax of expressions, derive 1\*(2-3+1) from e using rules defined above. Make sure your derivation satisfies the following rules:

(a) It must start from the start non-terminal.

(b) It must end with the input string

(c) Each step replaces a non-terminal with a new string

(d) Each step is guided by one production

b) Draw a parse tree for your derivation of 1\*(2-3+1). Your parse tree must be a tree that satisfies the following rules:

1. The root of the tree must be the start non-terminal
2. The leaves of the tree must be terminals
3. The internal nodes must be non-terminals
4. The parent and children relations must represent a production

c) Is the BNF ambiguous --- that is, is there an input for which two distinct parse trees can be built? If yes, give an example input that has two parse trees.

d) Try to simplify your parse tree into an abstract syntax tree (AST) representation.

2. For each of the given lambda terms, reduce it to normal form.

1. (λ x. x + y) y

2. (λ x. λ y. x + 5 \* y) y x

3. λ x . λ y . (λ z . z + 1) y

4. (λ x. (λ y. y x ) ( λ z. x z))(λ y. y y)

5. (λ x . λ y . x+ y) y z