**CSE420: Compiler Design**

**Assignment 01**

**Dead Line: 18/10/2017**

1. Consider the following grammar:

 S-> SS+

 S-> SS\*

 S-> a

* Do this grammar need left-factoring? If yes, left factor this grammar. [2]
* After step 1, do this grammar have left recursion? If yes, remove left recursion. [2]
* After step 2, determine the First set for the grammar. [2]
* After step 3, determine the Follow set for the grammar. [4]

1. Consider the following grammar, ***G = {bexpr, {bexpr,bterm, bfactor}, {not, or, and, (, ), true, false}, P}***.

*bexpr → bexpr or bterm | bterm*

*bterm → bterm and bfactor | bfactor*

*bfactor → not bfactor | ( bexpr ) | true | false*

* Do this grammar need left-factoring? If yes, left factor this grammar. [2]
* After step 1, do this grammar have left recursion? If yes, remove left recursion. [2]
* After step 2, determine the First set for the grammar. [2]
* After step 3, determine the Follow set for the grammar. [4]
* Generate LLparsing table for this grammar. [5]
* Parse the string “*not(false or (true and false))”.* [5]