# START OF QUIZ
# Student ID:
# 38595138,Christilaw,Tim

## Question 1

Topic: Lecture 3
Source: Lecture 3

Explain why the following rule is not valid in a CFG: NP VB -> DT NN VB (1)

## Question 2

Topic: Lecture 2
Source: Lecture 2

What properties of English syntax make regular expressions suitable for chunking? Do you think that this functionality would extend to many other languages? Briefly explain. (1)

## Question 3

Topic: Lecture 4
Source: Lecture 4

Imagine that you are a comedian writing jokes. How might you use an automatic parser to help you find material? Briefly explain. (1)

## Question 4

Topic: Lecture 1
Source: Lecture 1

Why does the substitution test work for identifying constituents? Do you think there are any constraints on what can be substituted? Explain briefly. (1)

## Question 5

Topic: Lecture 4
Source: Lecture 4

Why do we not evaluate parsers by the number of correct nodes in the tree? (1)

## Question 6

Topic: Lecture 2
Source: Lecture 2

Do you think that we could do dependency parsing and a constituency-based task (such as chunking) at the same time? What features of the tasks might support each other (additive qualities), and which might make such a task more difficult (adversarial qualities)? (2)

## Question 7

Topic: Lecture 3
Source: Lecture 3

Post-positive adjectives are adjectives that occur after the noun phrase they are modifying (such as "attorney/surgeon general", "somewhere nice", "nothing important"). Given that they tend to occur in set phrases, do you think it would be better to write a general class of PostAdj, and create PostAdj phrases in a CFG, or just list them as valid NPs (ie, NP = surgeon general)? Discuss the pros and cons of either decision. (2)

## Question 8

Topic: Lecture 1
Source: Lecture 1

Imagine someone develops a new parser that has 100% accuracy. The developer claims it has 100% on every test set they've tried. Why might you be sceptical of such claims? How would you go about trying to disprove them? (2)

## Question 9

Topic: Long
Source: Lecture 4

Imagine you're building a tool to help second language learners of language X. You have a grammar of their first language (L1), and a grammar of the language they are trying to learn (X). How might you build a tool that learns how to translate a production from L1 into X? Describe any additional data or tools you might need, and the process you would use to learn a "production-translation grammar". Also explain how you could use this to create illustrative examples of how the syntax of language X works.

# END OF QUIZ