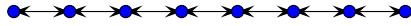# Grammar Module

Version 1

Introduction

This is your first module for your Designer Programming Language assignment.

Your task is to design a *grammar* for your very own, personal programming language. First, we need to understand grammars, so read *this*.

Specifics

Specifically, you are to write a complete grammar for a programming language of your own design. Store your grammar in a text file named *grammar.txt*

Your grammar should consist of approximately 25 rules.

You should also write a program in your language that illustrates every rule in your grammar. The program does not need to make any sense; it just needs to show the syntax of your language. Store your sample program in a file named *program.txt*.

Constraints

Your grammar must implement a dynamically-typed language. That is, type errors will be detected during program execution, not parsing. Failure to adhere to this constraint will result in a half-point deduction off your final overall percentage in the class.

Submission

To submit assignments, you need to install the *submit* system.

- *linux, windows 10 bash*
- *mac instructions*

You will hand in (electronically) your code for final testing with the command:

```
submit proglan lusth grammar
```

Make sure you are in your assignment directory. Only material in the top level of this directory will be graded. The submit program will bundle up all the files in your current directory and ship them to me. Thus it is very important that only the source code and any testing files be in your directory. This includes subdirectories as well since all the files in any subdirectories will also be shipped to me. You may submit as many times as you want before the deadline; new submissions replace old submissions. Old submissions are stored and can be used for backup. Clean out any large files/object files/executables in the directory or subdirectory before submitting,

You will need a password in order to submit. It will be given in class. Do not ask for the password on any public forum.