# Programming Languages - Final Project

## Groups

Groups of size 1-4 are allowed.

## Design a Language

For this project you will design a language. You are not designing C, Java, Lox, etc. but you may be influenced by existing languages. Your language must have a “feature” – some reason for existence – a solution to some problem.

If in doubt, ask me. I'm pretty much ok with all kinds of ideas. Some are less useful than fun.

## Suggestions:

**DSL**: An excellent notion is to design a domain specific language. A language designed for a particular domain that helps solve or write programs in a domain.

**General Purpose Language**: You can design a general-purpose language. What feature would you introduce to make programming languages safer, easier to use, easier to read, easier to write, or any other improvement.

## Grading Guidelines.

As for scoring it may vary as I expect languages to come in a range of complexities. So the weights here are not absolute but ranges. My goal in providing the ranged weights allows implementation-free projects without automatic failure (-10%) and the ability to reward implementations (+30%). Think of the first number as minimum deduction. The second number as maximum potential, allowing you to concentrate on your strengths.

X% - Y% == No less than X% but no more than Y%

### Presentation 20-30%

A great presentation needs to be your primary focus. Don't just hand me a zip with dozens of files. Help me out. And make it look great.

### Justification 10-15%

What is the reason for your language. Convince me it is needed and that it solves a problem. It doesn't have to be a great problem. But tell me your motivation for doing this beyond "it was assigned."

### Grammar 20-25%

Present the syntax and lexicon of your language in a readable form. This should also be part of the presentation but will be scored for its own value and merit. This is part of the instruction manual.

BNF, Railroad Diagrams, EBNF forms are allowed. It is reasonable that you may have two grammars, one for the implementation and one for presentation, but they should be aligned.

### Examples: 15-25%

Provide substantial examples of the language that showcase the capabilities/features.

### Working Implementation/Prototype: 10-30%

It is the intention of this project to provide room for invention. As such I believe it may be possible that design complexity may possibly forbid a complete implementation due to time constraints. For smaller, simpler languages the value/requirement of the implementation is more obvious. Make every attempt to provide a working interpreter (translating to an existing language is acceptable).