Omkar Prabhune, CS-C 49
11910471

So what is Dope? / Aim

To build a dynamically typed, interpreted and object oriented language. The interpreter itself will be built in Java, with a tree-walk design

Dynamically Typed

Dope is dynamically typed. Variables can store values of any type, and a single variable can even store values of different types at different times.

If you try to perform an operation on values of the wrong type — Say, dividing a number by a string, then the error is detected and reported at runtime.

Automatically Managed

High-level languages exist to eliminate error-prone, low-level drudgery, and what could be more tedious than manually managing the allocation and freeing of storage?

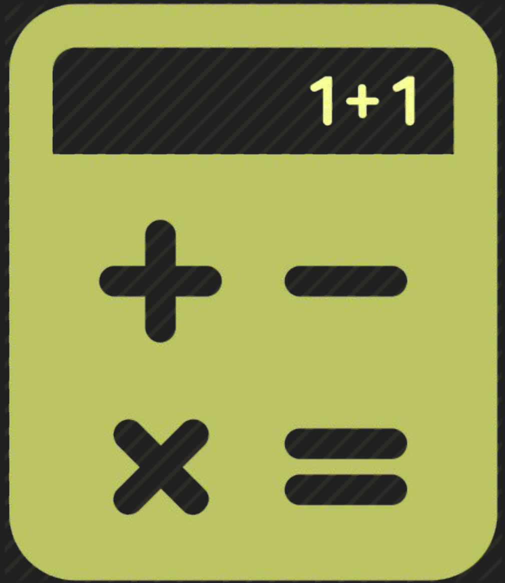
So, Dope will be using a custom Garbage Collector built and implemented in Java.

Data Types

- Number
- Boolean
- String
- Null

Expressions

1. Arithmetic (+, -, *, /)
2. Comparison (==, <, >)
3. Logical (!, and, or)



Precedence is the same as C-style languages

Object Oriented

Apart from the rest (statements, control flow and functions), Dope is also an object oriented language implemented with prototypes , allowing the use of classes and objects with basic OOP concepts such as:

- Encapsulation
- Polymorphism
- Inheritance

Literature Review

There’s quite a lot of existing programming languages out there, from old favourites like C to smaller, ‘indie’ ones like Elixir (A Ruby/Erlang hybrid).

After reviewing their documentation and working, Dope has been designed to provide as many features as possible, with the skillset of a CompSci Sophomore

References

1. David F. Bacon, Perry Cheng, V.T. Rajan, “A Unified Theory of Garbage Collection”, IBM Research

2. William Wold, FreeCodeCamp, “The programming Language Pipeline” https://www.freecodecamp.org/news/the-programming-language-pipeline-91d3f449c919/

3. Crafting Interpreters, http://www.craftinginterpreters.com/