

# Design Document

## 1. Introduction

- The language is called SEA++
- The language's Paradigm:
  - Stack-based language
- The Language's Unique Features:
  - String operations
  - Static type system
  - Infix to RPN notation of integer expressions

## 2. Design

- The language's Base Features:
  - Basic data types and operations will be implemented at the **core** level, because they are the foundation of our programming language.
  - Conditionals will be implemented at the **core** level due to the need for logic switching.
  - Recursion will be implemented at the **core** level due to the necessity of repetition in more abstract functions.
  - Stack manipulation operations will be implemented at the **core** level, because the operations will be defined as a part of a command
  - Procedures/functions with arguments will be implemented at the **core** level, because they will be a part of the basic command implementation.
  - Static type system will be implemented at the **core** level, because the type checking will be built with the basic data types.
  - Strings and operations will be implemented at the **syntactic sugar** level, because strings will be comprised of the basic data type "char".
- The language's Safety Systems:
  - Static Type System, no type errors
  - Stack will not underflow/overflow

## 3. Implementation

- The language's semantic domains:
  - Integers, boolean values, arithmetic expressions, strings, functions.
- Unique Implementation Highlights:
  - String concatenation and reversal.
  - Data types will be checked statically.
  - Reverse polish notation will be a unique feature to allow stack math.

peer connect:

atom://teletype/portal/3f72b351-0bbf-4df5-b5a7-e7afde3afc81