**Credence Labs**
**Project "Shvan"**

**Sunday, January 24, 2021**

**shvan_v1**

## A High Level Architectural Description Aspect

We intend to create a programming language which will have the following aspects:

- It will be an <u>interpreter as well as compiler</u>
- It will have a command line prompt for basic declarative <u>regular expressions</u> and mathematical operations (from arithmetic, algebra to basic calculus), in the initial phase
- It will definitely handle string operations such as concatenation operator, splitting etc. from command line as well
- It will have an IDE too
- It will also have <u>a special feature that programmer can add his own functionality as a library and api</u> <u>within their prompt</u> which will mimic MS Excel type macro or C/C++ style header cum source codes or Java type interfaces
- It will also have Java type interfaces for implementation left to the end programmer (this term is akin to "end user")
- It's source will be saved as .txt file and the compiled or interpreted by our linker
- It will incorporate some standard language constructs / capabilities such as
    - OS internal explorations, services
    - Networking & Inter-process services
    - Data Structures that can be used just like Java's container service or C++ STL or Python's List, Tuple, Dictionary
    - Batch processing
    - Binary Data Services like conversion of and to BIN, HEX, OCT or big/little endians
    - Primary Data types and also User Defined Data Types such as arrays or structures and classes
    - IO & Print to Printer or to console (very basic)
    - File Handling
    - MS Windows Specific Services
    - Linux Specific Services
    - Processor access (be it single, dual or multi cores) by assembler access. *Maybe we have to write a new assembler based on existing assemblers*

**Authors: Mr. B. Sundar and Mrs. Varsha Bandodkar**

**Credence Labs**
**Project "Shvan"**

- It will have some **special features** as follows:
  - Mathematical operations
    - New aspects will be declarative approach for finding logarithm (natural or other bases), basic calculus or financial aspects of Excel too
    - Basic Algebra like factorization or polynomial evaluations etc
  - Exclusive Mathematical Modules (XI/XII Science)
    - Scientific
    - Financial
    - Daily operations
  - Protocol implementers for hardware programming like
    - MODBUS
    - Virtual COM Port detection
    - Serial programming

- It will have the following features **eventually**:
  - Advanced Mathematical Operations such as
    - Advanced Calculus
    - Computational Physics
  - Image processing like OpenCV
  - Multithreading
  - Persistence
  - OOP fundamentals
    - Classes
    - Op Overloading
    - Function Overloading
    - Function Overriding
    - Friend & Virtual Functions
    - C++ types Templates
    - Inheritance
  - Database services
- It **will not be having** the following features (as of now):
  - No Internet programming aspects
    - No Client side
    - No Server side
  - No Graphics programming aspects
  - No UI creation aspects such as Tkinter or PyQt packages of Python
- It will have an exception/error handling that gives suggestions than point errors alone. It will have C++/Java style try…catch… or Python style try… except blocks for programmer
- It will be tested on Windows (7 & 10) and Linux (Ubuntu) of latest versions

**Authors: Mr. B. Sundar and Mrs. Varsha Bandodkar**

**Credence Labs**
**Project "Shvan"**

## Project Management Aspects

- We will develop a MS Excel based Project Management Workbook for list of actions, tasks, aspects to be done and completed
- Our source code, application logic, design document, code snippets, test codes, test cases, additional web related notes or references will be stored publicly in **github**
- Our project management workbook will be on Google Drive to track works
- We will be glad for people to be part of the project even now, provided people are those whom we can trust and have real value addition to the project than visitors and learners or freelancers
- In this project, there will be only contributors and no bosses. It is *of the techie, by the techie* and *for the techie* project that is iteratively grown by brainstorming sessions, though now it is only between two people

## Code of Conduct & Ethics

- We can borrow ideas, learn from books and references such as research papers or archaic contributions but **<u>we will never copy-paste or steal code from any other resources</u>**

References for us will be as follows:

- Languages - Python to make the interpreter, C & C++ to make the compiler, ASM for the assembly code to be embedded if needed to access process aspects
- Books - books on Compiler, Interpreter and of course the above language books that explore deeper aspects such as OS internals, Networking parts, Math functions etc
- Our own experiences and trials

**Authors: Mr. B. Sundar and Mrs. Varsha Bandodkar**