# Design of SOPA language:

**Instructions to write sample programs:**

* Each program should start with the keyword ‘start’ and end with the keyword ‘stop’.
* Variable names can be any alphabet, or sequence of alphabets, sequence of alphabets including digits and underscores.
* Variable names should start with an alphabet.
* Variable names cannot have any special characters except underscores.
* All declarations and commands should end with a semicolon.
* All declarations must be done before assignments and commands.
* Variables can only be assigned numeric, string, and Boolean values.
* For const variables, you can assign values directly but for var, you need to declare the variables first and then assign values.
* Sopa language supports nested for loops (nested repeat)
* Sopa language supports integer data types only.
* String assignments are allowed but string operations are not supported.

🡪 Declaration: The following are valid variable declarations:

1. var x;
2. const x = 5;
3. const x = "shailee";
4. const x = true;

🡪 Assignment: The assignment should be done in the following manner:

1. x := 8;
2. x := “shailee”;
3. x := true;

🡪 If-else statements: The Syntax is as follows:

if(condition)

commands;

else

commands;

**Note:** If there are multiple commands in the if statement then enclose the commands with start .. stop keywords.

🡪 Ternary Operator: The Syntax is as follows:

Variable:= condition? command if condition satisfied: command if condition not satisfied;

🡪 For Loops: The Syntax is as follows:

repeat(i,2,10)

show i ;

endrepeat

🡪 While Loops: The Syntax is as follows:

unless(condition)

command;

until

🡪 Square root: The Syntax is as follows:

sqrt(number).

**Important Notes:**

The language supports :

1. Operators and primitive types for Boolean values (and, or and not operators)
2. Numeric type (int)
3. Addition, Subtraction, Multiplication and Division, Modulus **(extra).**
4. Relational operators for numeric data types (greater than, less than, equal to, greater than equal to(extra), less than equal to **(extra)**)
5. String value assignments to variables.
6. Assignment operator.
7. A ternary operator (eg: ‘ ? : ’ in Java)
8. Traditional if-else construct.
9. Traditional for loop
10. Traditional while loop
11. Print statement
12. Square Root Functionality **(extra)**