

Milestone1 - SER 502 - Team 26

Team Members:

- Hardik Sakhuja (hsakhuja)
- Manisha Deshpande (mdeshp10)
- Sahithya Cherukuri (scheru20)
- Sameer Mungole (smungole)
- Saudamini Khare (skhare10)

Language Name: Saanp (.hiss) - Inspired by Python Programming Language

Language Design:

The language design defines a syntax for a programming language that includes a set of rules for constructing programs. The language design aims to provide a flexible and expressive syntax that allows programmers to write programs that can perform a wide range of tasks.

- Data Types:

- INTEGER
(A number is a sequence of digits that may be positive or negative.)
- STRING
(A string is a sequence of characters enclosed in double quotes or single quotes.)
- BOOLEAN
(True | False)

- Identifiers:

An ID is a variable name that starts with a lowercase letter and may contain lowercase letters, digits, or underscores. It cannot be a reserved keyword of the language.

- Operators:

For Boolean operators - [and, or, not]

For Integer operators - [+, -, *, /]

- Expression operators: An expression can be a comparison, a boolean operation, an arithmetic operation, or a value: precedence and associative operators.

- *Assignment Operators:*

- = operator is used to assigning a value to an identifier.

Identifiers: [a-z][a-z0-9_]* -

(True | False | if | else | endif | for | endfor | while | endwhile | range | print)

[An ID is a variable name that starts with a lowercase letter and may contain lowercase letters, digits, or underscores. It cannot be a reserved keyword of the language.]

- *Conditional Constructs:*

- Assigning value to an identifier from a ternary operation. (?:)
- If-else: An if-else statement checks a condition and executes a block of code if the condition is true; otherwise, it executes another block of code.

- *Looping Structures:*

- Traditional for loop: A for loop iterates over a range of numbers and executes a block of code for each number.
- While loop: A while loop executes a block of code repeatedly if a condition is true.
- Enhanced Loop: A while loop executes a block of code repeatedly as long as a condition is true.

- *Print:*

- A print statement outputs the value of a variable to the console.

Language Grammar:

```
PROG ::= BLK
BLK ::= STMT | STMT BLK
STMT ::= DEC | IFE | FOR | WHILE | EFOR | PRINT

DEC ::= ID = EXP. | ID = TER.

IFE ::= if BOOL: BLK endif | if BOOL: BLK else: BLK endif
TER ::= BOOL ? EXP : EXP

FOR ::= for ID = NUM, BOOL, INC: BLK endfor
INC ::= ID = ID + NUM | ID = ID - NUM | ID = ID * NUM | ID = ID / NUM

WHILE ::= while BOOL: BLK endwhile

EFOR ::= for ID in range(NUM, NUM): BLK endfor

PRINT ::= print(ID).

EXP ::= TERM | TERM + EXP | TERM - EXP | STR | BOOL
TERM ::= FACTOR | FACTOR * TERM | FACTOR / TERM
FACTOR ::= ID | NUM | (EXP)
BOOL ::= True | False | CMP and CMP | CMP or CMP | not BOOL | CMP
CMP ::= EXP == EXP | EXP != EXP | EXP < EXP | EXP > EXP | ID

ID ::= [a-z]+[a-z0-9_]*
STR ::= "[^"]*"
NUM ::= [0-9]+ | -[0-9]+
```

Language Information:

Interpreter:

- *Prolog*
- *Data Structures: List*

An interpreter is a computer program that reads and executes code in a programming language. It translates the source code into machine code, line by line, and immediately executes each line as it is translated. Interpreters are commonly used for scripting languages, prototyping, and rapid development due to their ease of use and flexibility.

Parsing Technique:

- *Prolog*
- *Data Structures: List*
- *Parsing Technique: Top-down*
- *Grammar: DCG*

Parsing is the process of analyzing a string of symbols according to formal grammar rules. It involves breaking down the input into its constituent parts, determining its structure and relationships, and generating a parse tree or abstract syntax tree that represents the meaning of the input.