**Syntax Design and Language Specifications of Dragon**

**Mut variables,** **let expression, and rest all things like var** , **const**, **data types** , **if , else, while loop, for loop in python,** Statically typed and dynamically typed language.

1. **Data types**

Our language will contain the following 4 data types: int, fraction, boolean and string.

**Int:** an integer from the range -1018 to + 1018.

**Fraction**: of the form x/y, truncate up to 6 decimal places.

**Bool:** True or false

**String:** series of characters. Eg: “hello world”, “dragon”, “123abc”.

1. **const keyword**

In dragon, the “const” keyword will be used to indicate that a variable or an object’s value cannot be altered, if tried to do so, the interpreter will generate an error.

Below is the syntax to declare a constant variable.

const int a = 5;

a = 6 // generates error as we tried to change the value of a constant variable

1. **var keyword**

The “var” keyword is used to declare variables. When the variable is declared used var, the interpreter will create a new variable in the current scope with the given name.

var x;

print(x); // undefined value

var can also be used to assign value at the time of declaration.

var x = 5;

print(x); // 5

In dragon, var will be function - scoped which means that the variable declared with ‘var’ is only accessible within the function it was declared in, and in any nested functions.

Eg:

function myFunction() {

var x = 5;

print(x); // 5

}

print(x); // ReferenceError: x is not defined

1. **Let keyword**

‘Let’ will be used to declare variables similar to ‘var’, but the key difference between let and var is that let is block scoped whereas var is function scoped.

Eg:

{

let x = 5;

print(x); // 5

}

print(x); // ReferenceError: x is not defined

Unlike var, when the interpreter will encounter a variable declaration with the ‘let’ keyword, it will create a new variable with the given name if a variable with the same name does not exist, else it will generate ‘syntax error’. Unlike const, let variable can be re-assigned.

1. **Some valid uses of let**

Let b = 2 in let a = 5 in a \* b;

Let a = 5 in let a = a + a in a \* a;

Let a = 5 in let b = a \* a in b \* b;

(let a = 5 in a \* a) \* (let a = 6 in a \* a);

Let a = 5 in Let b = 6 in

a \* b \* (let b = 7 in a + b) + (let a = 7 in a \* a + b);

1. **Mutable variables**

If the “mut” keyword is used after let, then the variable declared becomes mutable, otherwise not.

Eg:

Let mut p = 1 in {

Let mut i = 1 in {

While i <= 10 {

p = p \* i; // valid as p is mutable

i = i + 1; // valid as i is mutable

}

}

}

1. **If - else condition**

if (cond 1) {

expression 1;

expression 2;

}

else if (cond 2) {

expression 1;

expression 2;

}

else {  
 expression 1;

expression 2;

}

1. **While loop**

This is an entry controlled loop.

let i = 0;

while (i <= 10) {

print(i);

i = i + 1;

}

1. **For loop**

This is also an entry controlled loop, with an initialization expression, test condition and an update expression.

Eg:

for (let i = 0; i <= 10; i ++) {

print(i);

i = i + 1;

}

1. **Both dynamic and statically typed**

Our language dragon will support dynamic typing i.e. the type of the variable will be determined at the time of run time, and the type of the variable can be changed during the execution of the program.

Eg:

Let a = 5;

a = “hello”

The interpreter will automatically change the type of variable a from integer to string.

But if you define the type of the variable explicitly using any of the above four mentioned data types, then dynamic typing won’t be allowed, at that time, the language will act as statically typed.

Eg:

int x = 6;

x = 7.9; // error