LAB 02

ABDUL SUBBOOH DANYAL
ADVANCED PROGRAMMING
2/25/15

INERPRETER

In this lab I have implemented the interpreter using Java for an unknown language. The language initializes a variable with the keyword "Let". And for printing the lines and variables it uses keyword "print". And it also perform arithmetic operations quite efficiently. If a variable is not initialized and used in the code it will display an error message. I you initialize a variable twice it also give you an error. If you use "==" for declaration of a variable it also give you an error. It also maintain operator precedence in performing mathematic operations. The code of the interpreter is uploaded on the GitHub and here is the link for public repo https://github.com/12bscsadanyal/Interpreter.git. The following are the tests for testing my code.

Arithmetic Precedence:

The following snap shot shows that my code is maintaining operator precedence.

```
Get Some Arithematic
x = 2 + 2 * 2
4
After operations  4
After operations  6
```

Incorrect Initialization:

If you try to initialize a number with "==" it will give you an error. Below is the scree shot.

```
Declaring new variable
Let x == 4
wrong syntax
```

Correct Initialization

The correct initialization is shown below.

```
Declaring new variable
Let x = 4
{x=4, integer=26}
```

Multiple Declarations Not Allowed:

The multiple initialization of the same variable are not allowed in the interpreter.

As shown below.

```
Declaring new variable
{x=4}
Declaring new variable
{x=4, y=10}
Declaring new variable
Float is initialized
Declaring new variable
Multiple Declerations are not allowed.
```

Float is supported:

We can initialize a float with the number that containing "." In the value. Shown below.

```
Declaring new variable
Let z = 40.52
Float is initialized
{z=40.0, integer=26.25}
```

String is supported:

We can also initialize a variable with the type string. The string containing its value in the "Quotation notes".as shown below.

```
Declaring new variable
Let v = "Danyal"
String is initialized...
{v="Danyal"}
```

Incorrect Initialization is not allowed:

```
Declaring new variable
Let 0 = 8
wrong syntax
Declaring new variable
Let y = po
wrong syntax
```