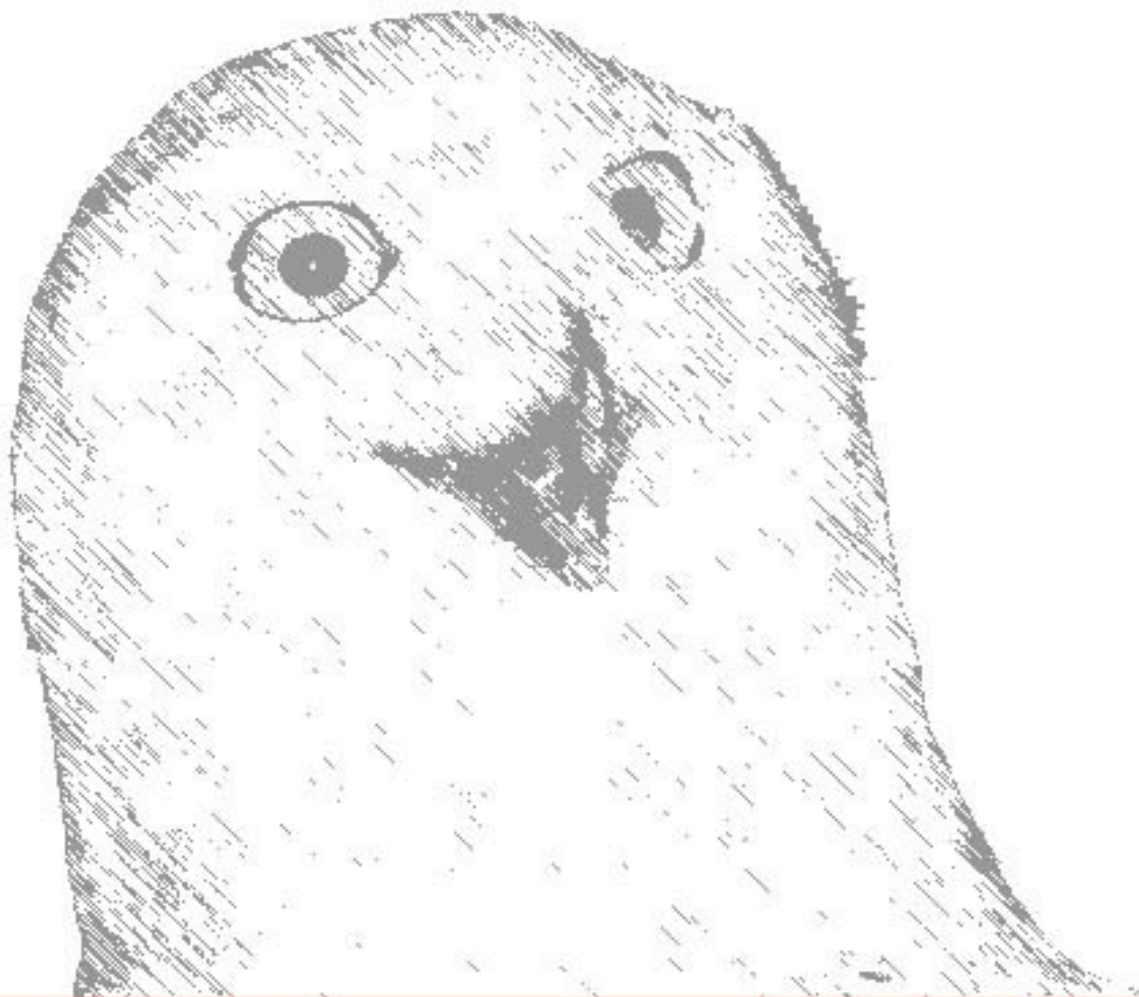


*Learning how to create programs
with ORLY?*



ORLY?

PROGRAMMING LANGUAGE

Programming Guide

BStreamZDevelopment[®]

Welcome to the **ORLY?** language programming guide. This esoteric programming language is decoded in Java. It was written in 2008 by William Reehil of *BStreamZ Development*.

What can be done in ORLY?

Like in many programming languages programmers can use conditionals, arithmetic, boolean values, and the input/output stream. **ORLY?** can use both the command prompt window and GUI components to retrieve and display data. **ORLY?** uses Java Swing components for the GUI interfacing. It cuts down on the Java code required for swing interfacing.

Syntax

This language is case sensitive. The programmer can use any variable from A-Z, all capitals. All variables are initialized to 0. Hard coded values and variables can be used interchangeably in the spaces between delimiters. (A delimiter is a character used to split up the code into segments)

The delimiter is |

eg:

5|5

Comments

??| Followed by comments are ignored and the next line is processed

eg:

??|This line is ignored in the main code

Declaration of Variables

The equal statement starts with a ~ followed by the value and the variable.

eg:

~|5|A (sets A equal to the integer 5)

~|B|A (set A equal to B)

Input/Output Stream

ORLY? is the command for output of the value to the right

eg:

ORLY?|5 (would print a 5)

Same goes with a variable (say A has a value of 6)

ORLY?|A (would print a 6)

SIN (abbreviation for System In) is the command to get input from the user through a Java Swing JOptionPane.

SIN|Enter Value for (variable)|(variable) The second place is for the programmer to write a string that will appear in the JOptionPane window. Whatever the user enters, the variable will receive that value, unless it is not a valid integer, in that case an error message will pop up and prompt the user to enter it again

UMESS (abbreviation for User Message) sends the user a string message to communicate with them

UMESS|Hello World! (This snippet of code will print Hello World)

EL is the command to end the current line

UMESS|Hello|EL

UMESS|World!

The above code would print:

Hello
World!

Arithmetic

Every mathematical equation is expressed first the operator then two operands and then the variable that receives the value

eg:

-|A|B|C (Subtraction: A minus B equals C)

+|A|B|C (Addition: A plus B equals C)

*|A|B|C (Multiplication: A times B equals C)

/|A|B|C (Division: A divided by B equals C)

^*|3|A|A (Raise to power $A=A*A*A$)

/#|2|81|A (The N root of the second number stored in the third, meaning $A=9$)

Conditional Statements

^ greater than

_ less than

^~ greater than or equal to

_~ less than or equal

~~ equal

X~ doesn't equal

The setup is as follows:

^|A|B|C means $A>B$ if it is true C gets a 1 otherwise it gets a 0

The IF statement

IF is the command for the beginning of an if statement, it is followed by a variable that has to be true for the code segment to be executed

EIF is the command for the end of an if statement
ex.

IF|(variable)
blah blah blah
EIF

Looping

L337 is the marker for the loop it has to be followed by a variable A-Z to identify it

@|(variable)|condition means if the condition is true it will loop back to the top of the loop

Within the loop the condition needs to be set or else the loop will continue uninterrupted

L337|Z Names loop as Z

+|C|1|C Adds 1 to C

ORLY?|C Prints C

_~|C|20|D $C \leq 20$ stored in D as either 1 or 0

@|Z|D Loop back to Z until $C > 20$

GUI "Bill's Programming Language Compiler"

The layout is pretty straight forward, it is a basic text editor to compile code written in my language. It has a few buttons that have different functions. It also has some Menu bar items. There is a button that saves, compiles, and runs. This button does this by saving the text in the JTextArea to a file named "new" in a precoded directory. It then compiles the Test.java file with the text from new.txt as input. The way it does this is the start of a separate process that calls "javac -cp Test". The output from Test.java is then called upon in a new frame from the run button. The separate save button does the same as the save function in the "all in one" button, except the programmer is prompted for a name to give to the file. The separate compile button compiles the file the same as the "all in one" compile but it also displays a success message. Run is the same. The Menu items call the exact same function as their counterparts. The new menu item clears the text area. The edit functions do the same as they would in a text editor, copy, cut, paste, and select all. Help just shows some syntax help. About shows the name of the program, version, and year created.

Example Programs

Code:

```
??|This Program displays the 2, 3, and 4 time tables|
~|2|X
~|2|Y
L337|A
ORLY?|X
UMESS| times table is as follows:|EL
~|2|F
L337|B
*|X|F|D
ORLY?|D|EL
+|F|1|F
_~|F|12|E
@|B|E
+|X|1|X
_~|X|4|V
@|A|V
```

Output:

```
2 times table is as follows:
4
6
8
10
12
14
16
18
20
22
24
3 times table is as follows:
6
9
12
15
18
21
24
27
30
33
36
```

```
4 times table is as follows:
8
12
16
20
24
28
32
36
40
44
48
```