Manual for the 41++Language

Kavi Gupta

June 13, 2015

1 The Concept of Programming in 41++

41++ is a language that allows people to talk to computers in a language any human should be able to understand. The language uses simple words and phrases to describe what are, in fact, simple concepts that many languages confuse with symbols. 41++ is *not* the most useful language on the planet for writing airplane control systems, payroll distribution systems, or other programming languages (although we *will* implement a compiler in the future to demonstrate some of 41++'s features). Instead, 41++ is intended to be a programming language so that beginners can learn the basics of algorithms, functions, and structures in a text-based yet simple format.[1]

2 The Console

This section will make a lot more sense if you have 41++ Editor open, so I suggest you open it now.

The console is the little box at the bottom, marked, as you might suspect, "41++ Console." You can write stuff into it and press "ENTER". The usefullness of the console will be discussed later in this section.

3 Literals

Try typing 2 into your console. You should get something that looks like this:

```
>>  2
 =  2
```

Notice how the second line is in yellow; this, along with the equals sign before it, signifies that it is an output.

Now, you may be thinking "OK, that's great, it's a duplication program. Very very useful. [end sarcasm]", but in fact explaining this simple characteristic will give you some insight into how to use 41++. When you enter 2 into the document, then the 41++ interpreter calculates the literal value of the input, which in this case is helpfully 2.

Anything that when typed into the console outputs itself is known as a "literal". Numbers are an example of such a literal. Valid numbers consist of a sequence of digits 0-9, with an optional period that must be followed by more digits[2]

The second type of literals are strings. Strings are slightly more complicated than numbers, but the simple rule is that they must start and end with a single quote ('). Here are a few examples:

```
>>  'abc'
 =  'abc'
```

[1] If you don't know what any of that just meant, don't worry—just read the rest of this manual to find out!

[2] Some languages allow for numbers like 4., but periods have a special meaning in 41++ so you must write 4.0 or just 4 (no period at the end).

```
 >>   '1 2 3'
  =   '1 2 3'
```

```
 >>   'In \'quotes\''
  =   'In 'quotes''
```