Algonquin College Logo

# SCHOOL OF ADVANCED TECHNOLOGY

### ICT - Applications & Programming

### Computer Engineering Technology – Computing Science



A11

Language Specification

Team:

[Joshua Fearnall] - Id: [041019251]

Language Name [Garter]

***This template is suggested (not mandatory) to answer A11 Specification.***

|  |  |
| --- | --- |
| **Part**  **1** | **Language User Reference** |

**EXPLANATION**

*The purpose of this assignment is to invent a new computer language.*

* *This language can have the syntax and structure of your choosing.*
* *Option 1: Adapt the ‘BOA’ language to be Python compatible.*
* *Option 2: Define a DSL – Proper to solve specific problems (ex: science, economy, music, etc.)..*
* *This is going to be a fairly basic language. There's a lot of functionality that we'll be skipping over, while we implement the basics. You will need to tell me those basics, of course. In this document, I'm going to explain the steps of what to do with a bit of detail.*
  1. **User Manual**

**Element 1: Name / Extension**

*[Name your language! Name it after a city that means something to you. We suggest you use one "word" for the name.]*

*[What is the filename extension of your language? For example, for C it is .c, and for Professor Paulo's* ***Boa language*** *it is ".****boa****".]*

*[What is your language patterned after, or what is it similar to? What languages are inspiring your choice? It's okay if you're following Python closely.]*

My language is named Garter, file extension .gar

It will be like python (which might be a challenge because I don’t know python)

**Element 2 – Comments**

*[Comments: I want to do comments in your language. How do I write them?]*

Comments are written by placing the hash(#) character on a line. All characters written after the # are part of the comment until we reach a new line.

**Element 3 – Keywords**

*[Keywords: List the sequence of reserved / key words from your language]*

and, def, elif, else, False, for, if, import, in, not, or, return, True. none

**Element 4 – Variables and Datatypes**

*[Datatypes: Define integers, real numbers (float points) and strings]*

* *How many bytes are you needing for your variables? This determines their ranges. (Chambly, for instance, has a special 64-byte integer. This is ridiculously huge for most purposes.)*

*[Remember to define the number of bytes – and, if possible, range]*

int – 24 [ -8388608 to 8388607 ]

float – 24 [ -2.2250738585072014e-308 to 1.7976931348623157e+308 ]

string - 49 ( + 1 byte per additional character)

**Element 5 – Variables and Datatypes**

*[Variables: How would a programmer define variables that can hold integer numbers (numbers with no decimal point), floating point numbers (numbers with a decimal point) or text (ie: strings in Java). This is element 1. Consider if you want to flag the variables in a special way, like SOFIA or BASIC, or not, like C or Java.]*

Integers are defined by assigning a numeric value without a decimal point to a variable

Floats are defined by assigning a numeric value with a decimal point to a variable

Strings are defined by assigning a value surrounded by quotation marks to a variable

**Element 6 - Commands**

* ***Attribution****: How does your language let a programmer assign a value to a variable? (Will you allow casting? If so, how will it work?) How will your language handle math, and will it allow strings to be concatenated (merged)?*
* ***Selection****: How does your language do if-style logic? (Optional: Do you want to do some kind of switch/case as well?). You will need to explain how "conditionals" work in your language. How do you write Boolean operations, such as "or", "and", "not", and other conditions, such as less than, greater than, etc?*
* ***Interaction****: How will your code handle looping? (You can do one or more of a for-style loop, a while/do loop, etc.)*
* ***Input****: How does your program get input from the keyboard? (Strings are easiest.)*
* ***Output****: What would a programmer type to put output on the screen? What sort of variables or data will your code take?*
* ***Functions****: [Function definition: parameters and returning types]*
  + *What will be the syntax for making a function or subroutine?*
  + *How will it take parameters?*
  + *How will it return results?*

**Attribution**: The main variable types are assigned using [ varName = varValue ]

Casting is handled explicitly using functions of int, str, and float, e.g. (int to string)

x = 1

castedString = str(x)

Basic arithmetic will be handled using operators: ( +, -, \*, /, %, and ^ ) any additional mathematics functions would be handled by a separate library

String concatenation will be accomplished using the + operator with string literals or variables

**Selection**:

If statements work as follows:

if (x > 2)

Do()

elif

Do()

.. (as many elif as you want

else

Do()

Boolean operations will be written as follows:

If x OP y:

Foo()

Where OP is one of the following: [ and, or, not, <, >, <=, >=, ==]

Certain operators can only be used with specific datatypes, <, >, <=, >= are only used with numeric data

**Interaction:**

Looping: a basic loop will look like the following.

for name in collection:

Do()

**Input:**

Input is collected using an input() function; after the input function is called it waits for the user to enter an input and returns it.

**Output:**

Output is displayed using the print() function; after the function is called, the variable supplied as a parameter is displayed on the console. It will take any type of variable and convert it to a string.

**Functions:**

The syntax for a function is:

def functionName(parameters):

doSomethingHere()

Optionally the function can contain a return statement to return the results.

e.g. return something()

**Element 7 – Proper elements**

*[Include specific features / elements to be included in your language]*

* *What you could include / modify? Think about new datatypes / structures / commands, etc.*
* *Note: Do not share this info (it is supposed to be your proper elements in the language.*

I would like to try to add a ternary operator to my language, with syntax like C#. Where when the condition is true the first expression is evaluated, otherwise, the second expression is evaluated

conditionalExpression ? firstExpression : secondExpression

e.g.

# this would set the value of v1 to 0

v1 = (1 > 2) ? 3 : 0

|  |  |
| --- | --- |
| **Part**  **2** | **Examples** |

**Option 1: Python-like**

**Hello World**

|  |  |  |
| --- | --- | --- |
|  | def main():  print(“Hello World”)  main() |  |

**Sphere Volume Expression (or any other example)**

|  |  |  |
| --- | --- | --- |
|  | def volume():  pi = 3.14  r = float(input(‘Radius of sphere: ‘))  volume = (4/3) \* pi \* (r^3)  print(‘Volume: ‘, volume)  volume() |  |

*[TIP: See examples in the Lecture Notes –* ***Appendix 1****]*

**Option 2: DSL**

**[Your example here]**

|  |  |  |
| --- | --- | --- |
|  | [Your Code here] |  |

|  |  |
| --- | --- |
| **Part**  **3** | **Architectural Aspects** |

**Advantages**

*[What's the goal of your language? Are you trying to make something simple, fun, complicated? My personal language, Chambly, is based around being useful to scientists. (You can just make something up here, honestly. Think about it a little bit, have a little fun.)]*

The goal of my language is just to include the barebones of the python language. I don’t want to create anything too complex, and since it’s python-like I don’t want to deviate too much from the original either.

**Strategy: C Implementation**

*[How your language can be implemented in C – ex: datatypes]*

* *In plain English, or maybe even some high level pseudocode, how are you going to parse your language? You will be writing a compiler for your language, so these are some things you need to think about.*

The idea is we will go through each line of the computer program line by line, char by char, saving each character as it goes along. When it finds certain characters (like an empty space) it will take the characters it is currently holding and store them somewhere. Once an entire line has been read the input will be taken and executed according to the languages grammar rules.

*[Your ideas about how to identify elements from language]*

* *Consider your "write to the console" command as an example. How will your compiler detect it? How will it sort out what to write to the console? What if there's some literal text (ie: "this is going to get printed") instead of variables?*

When the compiler detects a string of characters with brackets that is not preceded by a return type ( e.g. print() ) the compiler will know it’s found a function. If the print function was provided a string variable, it will print the contents of that variable; whereas if the argument is a string literal the compiler will convert the literal to a temporary variable, then print it.

*[Your ideas about how to identify scope (ex: blocks between conditionals or functions)]*

* *How do you mark a block of code? If I use your loop logic, how do I control what portion of code gets looped through? In C, you might use { and }. In Python, the indentation is what matters. How does it work in your language?*

As in python, blocks will be determined by indentation

**Basic ideas about C implementation**

*[Which structures or datatypes you imagine to use in your language implementation]*

* *What do you think is going to be really hard about this? What would be, in your opinion, the hardest part of parsing your own new language? You don't have to write an essay, a paragraph or two will be fine.*

I think I’ll be using a tree for syntax analysis, a dictionary or hash table for the symbol-table, and maybe a stack during lexical analysis. Just getting my head around all these data structures (as well as the ones I’ve forgotten) is going to be challenging on its own.

Figuring out different strategies for handling things like how to tell the difference between a variable that has been initialized, and one that’s not, matching brackets, etc. Everything seems challenging.

***Note 1: C Datatypes***

*Remember that you are implementing your language in ANSI C. For this reason, you cannot create arbitrarily your language (from scratch). You need to use what is already provided by C Compiler. For this reason, think about using and defining the language obeying the datatypes.*

**Problems when using C implementation**

*[Your vision about main problems / difficulties when implementing a new language (ex: memory allocation, range of datatypes]*

Honestly it will all be a challenge. As I went through this document, I lost count of the number of times I had to stop to google a term or review something from the text.

From my basic understanding of what I need to accomplish to create a compiler I think memory allocation might be what I struggle with the most.

**FINAL SUGGESTIONS**

*Here some ideas to think about your language....*

* *Don't make this assignment harder than it needs to be on yourself. Focus on making the syntax for your language that meets our requirements. Worry about extra features later.*
* *Don’t worry if your new language winds up having really difficult parts. You'll be allowed to change your language as you go along, as long as you make "patch notes" to explain those changes. We'll tell you about this later.*
* *There's a marking key at the end of* ***CST8152\_Compilers\_F22-A11-Specification*** *that should steer you along for grades. Focus your efforts on where you'll get the best results.*
* *Finally, think about creating an “master-piece”: until now, you have used several languages. And if you have conditions to define yours, how it could be?*

**References**

*[Include eventual references used here]*

Algonquin College

Fall, 2022