Algonquin College Logo

# SCHOOL OF ADVANCED TECHNOLOGY

### ICT - Applications & Programming

### Computer Engineering Technology – Computing Science



A11

Language Specification

[Quoc Thang Tran] - Id: [041009239]

Language Name [Krait]

***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.] Krait*

*[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****".] .krait*

*[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.] Python*

**Element 2 – Comments**

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

*# At the beginning of the line, the whole line will be considered as a comment (single-line)*

*“““ content “““ Use multiple line strings and # at the beginning of each line (multiple-line)*

**Element 3 – Keywords**

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

*False, True, and, or, not, continue, break, pass, if, elif, else, for, while, global, in, return, def.*

**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]*

*Text Type: str – 1 byte per char*

*Numeric Types:*

* *int – Integer – 4 bytes ( range from 231 to 231-1)*
* *float – Floating number – 4 bytes ( range from -3.4E38 to 3.4E38)*

**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.]*

*Define a variable:*

* *x = 1 (Integer)*
* *x = 1.5 (Floating number)*
* *x = “Hello” or x = ‘Hello’ (String)*

*Define a constant: (Mostly is defined in a separated file)*

* *NAME = “Brian” (String)*
* *AGE = 20 (Integer)*
* *Height = 14.5 (Floating)*

**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)?*
  + ***Assign a value to a variable****: (variable) = (value)*
  + ***Casting****:*
    - *int -> float: x = float(1)*
    - *float -> int: x = int(1.5)*
  + ***Math:***
    - *x + y (addition)*
    - *x – y (subtraction)*
    - *x \* y (multiplication)*
    - *x / y (division)*
    - *x \*\* y (exponentiation)*
    - *x % y (Modulus)*
  + ***String concatenation:*** 
    - *x = “Hello ” + “World” (x = Hello World) or*
    - *print(“Hello ”, “World”) OUTPUT: Hello World*
* ***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?*
  + ***if, elif (else if), else logic***
    - *if a>b (return TRUE If a>b)*

*…*

* + - *elif a=b (if the above condition is not true, try this condition)*

*…*

* + - *else ( if the above two conditions above are not true, run the code block below this else)*

*…*

* + ***and, or, not***
    - *a > b and a>c ( if a is greater than both b and c, )*
    - *a > b or a>c (if a is greater than b or a is greater than c)*
    - *not(a < b) (if a is not smaller than b)*
* ***Interaction****: How will your code handle looping? (You can do one or more of a for-style loop, a while/do loop, etc.)*
  + ***while loop***
    - *while (x<10):*

*x = x + 1*

* + ***for loop***
    - *for x in [5, 6, 7]*

*pass*

* ***Input****: How does your program get input from the keyboard? (Strings are easiest.)*
  + *input(“Enter you name: “) ( input is the function name and the remaining is the message)*
* ***Output****: What would a programmer type to put output on the screen? What sort of variables or data will your code take?*
  + ***Output on the screen:*** *print(output)*
  + ***Types of data for output***
    - *int*
    - *float*
    - *str*
* ***Functions****: [Function definition: parameters and returning types]*
  + *What will be the syntax for making a function or subroutine?*
    - *functionName()*
  + *How will it take parameters?*
    - *functionName(Parameters)*
  + *How will it return results?*
    - *functionName(Parameters):*

*x = 10*

*return x (return x as the result of the function)*

* + *Function call?*
    - *functionName(Arguments)*

**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.*
  + ***switch*** *(New feature compare to Python) – Replacement for long and nested if-else logic*
    - *switch (value):*

*case(1st case):*

*…*

*break*

*case (2nd case):*

*…*

*break*

*…*

*finally:*

*…*

* *Note: Do not share this info (it is supposed to be your proper elements in the language.*

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

**Option 1: Python-like**

**Hello World**

|  |  |  |
| --- | --- | --- |
|  | [Your Code here]  helloWord():  print(“Hello World!”)  helloWorld() |  |

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

|  |  |  |
| --- | --- | --- |
|  | [Your Code here]  volume():  pi = 3.1415926535897931  radian = input(“Please enter the radian: “)  volume = (4/3) \* (pi \* (radian \*\* 3))  print(“The volume is: “, 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.)]*

***I simply want to create a simple language and have fun while creating it.***

**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.*
  + *Datatypes*
    - *int -> integer*
    - *float -> floating number*
    - *str -> string*
  + *Function definition: def -> definition*

*[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?*
  + *The compiler will detect the a function’s name followed by 2 parentheses and maybe with arguments inside as a function call.*
* *How will it sort out what to write to the console?*
  + *The argument will be put within the two parentheses after the function’s name.*
* *What if there's some literal text (ie: "this is going to get printed") instead of variables?*
  + *Put two double-quotes, one at the beginning and one at the end of the literal text and put them all inside the two parentheses*

*[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?*
  + *I will use <tab> as the indentation. And : as the start of a code block*

**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.*
  + *The hardest part for me is to keep every rules and syntaxes consistent. As a new language (Python-like), it would be a challenge to follow all the new syntaxes and features strictly.*

***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]*

*Keep track of memory, avoid go beyond the range of datatypes and structure the code blocks properly are main difficulties in my opinion.*

**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