Algonquin College Logo

# SCHOOL OF ADVANCED TECHNOLOGY

### ICT - Applications & Programming

### Computer Engineering Technology – Computing Science



A11

Language Specification

Team:

[Student Name] - Id: [Student Id] / [Student Name] - Id: [Student Id]

Language Name [City / Country / Place]

***This template is suggested 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. It's fine if you pattern it to follow the SOFIA example closely, but you may imitate other languages, or invent your own, as long as you're clear about the origins of what you're doing.*
* *Your choices are not set in stone. You may make changes to your language as we go along, as long as you make "patch notes" explaining what those changes are.*
* *Bear in mind that the purpose of this course is to create a compiler that will compile your language. So, I suggest not making things more complicated than they need to be.*
* *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. **General Language Overview**

**Name**

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

**Extension**

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

**Genealogy**

*[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 SOFIA closely.]*

**Advantages**

*[What's the goal of your language? Are you trying to make something simple, fun, bedevillingly 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.)]*

**Hello World Example**

*[Create a basic Hello World Example here]*

* 1. **User Manual**

**Element 0 – Comments / Keywords**

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

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

**Element 1 – 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]*

*[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.]*

**Element 2 - Commands**

* *Assignment: 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?*

**Element 3 - Modules**

*[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?*

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

**Hello World**

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

**Sphere Volume Expression**

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

*[TIP: See examples in the Lecture Notes – Appendix 4]*

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

**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.*

*[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?*

*[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?*

**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.*

***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]*

*BONUS:*

* *You've identified some difficult parts. What are your thoughts on how you might solve them?*

**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\_A11Spec\_S21 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

Winter, 2022