**PLP-2 PROGRAMMING LANGUAGES - CS 330**

***FORTRAN***

Mars DiPillo

1. Does your language have keywords or reserved words? How many?
   1. Fortran has keywords and no reserved words (and those keywords are NOT the reserved words). It depends on the type/version of Fortran of how many keywords there are… though in the most recent type, Fortran 2018, there are over a hundred keywords.
2. What are the naming requirements for variables in your language?
   1. For variables, usually they need to be named specifically on what type of information is going to be stored within. Some examples of types available in Fortran are but not limited to; INTEGER, REAL, and LOGICAL. There appears to be no super specific types for numbers, if it’s an integer, use INTEGER, but any other types, granted they are real numbers, use REAL. Furthermore, to name a variable, you would have to type it as such; INTEGER : : y = 1. Note the usage of two colons between the data type and the variable name, then the variable name has an equal sign before the number.
3. Is your language statically or dynamically typed?
   1. Fortran is a statically typed language.
4. Strongly typed or weakly typed?
   1. Fortran is a strongly typed language.
5. Are some variables mutable while others are immutable?
   1. Discovering which variables are mutable or not was never clearly defined in Fortran. It has been widely discussed within the community but it appears nothing is entirely set in stone. Based on how Fortran is a compiled language, theoretically, every variable is mutable at some point. It also seems that the variable type is the only truly real immutable thing in Fortran.
6. What are the operators available for each data type?
   1. The operators in Fortran are \*\* for exponentials, \* for multiplications, + for addition, - for subtraction, and / for division. There are the additional logical operators, in two aspects. == for equality, /= for inequality, >, < greater than or lesser than operators, >= and <= for greater than or lesser than equal to operators. The second group of operators are .and., .or., .not. , .eqv. , .neqv.
7. Are mixed type operations allowed? If so, how are they accommodated?
   1. Mixed type operators from my experimentation with coding in Fortran, it seems to allow mixed operators with integer and real numbers. This language seems to accommodate this to happen by having the result either in a real or complex, depending on which you use.
8. At what point are identifier names and operator symbols bound in your language?
   1. Upon research, there is no exact, clear answer for when identifier names and operator symbols are bound in Fortran. However, from best judgment and research, it seems that identifier names are bound before the program is compiled.
9. For example if you declare a (variable, class name, function name), when is it bound to the type, address? When are operators (+,\*, etc.) bound to their operations?
   1. Declaring a variable, it will be bound to the type and operators when it is compiled in Fortran.
10. Explicitly typed or implicitly typed?
    1. Fortran is actually both (historically implicit)! You can choose whether or not you want your variables to be implicit or explicit. You can inform the compiler that you want explicit variables by typing ‘implicit none’ at the top of your program.
11. What about naming conventions? Are those enforced by the compiler/interpreter, or just standards in the community?
    1. For naming conventions in Fortran, they are enforced by preferences overall. Though based on the standard library official language, it’s preferred to use all lowercase for anything. Based on Fortran’s purpose in its long history, naming conventions were not a huge priority.

Sources:

<https://ourcodingclub.github.io/tutorials/fortran-intro/>

<https://fortranwiki.org/fortran/show/Keywords>

<https://courses.cs.washington.edu/courses/cse341/00sp/general/types.pdf>

<https://www.baeldung.com/cs/statically-vs-dynamically-typed-languages>

<https://fortran-lang.org/learn/quickstart/variables/>

<https://fortran-lang.org/learn/best_practices/style_guide/>

<https://okmij.org/ftp/meta-programming/mutable-var.html#:~:text=In%20Fortran%2FAlgol%20tradition%2C%20the,variables%20can%20be%20accessed%20directly>.

<https://web.chem.ox.ac.uk/fortran/arithmetic.html>

<https://fortran-lang.org/learn/quickstart/variables/>

*The reason why there are a few personal websites or wiki’s is because, shockingly, there isn’t so much official documentation on Fortran on the web. There is the official Modern Fortran website, but that only went so far.*

*Verdana font used for its readability for those with dyslexia. Accessibility!*