CCS314-18 THEORY OF PROGRAMING LANGUAGES

ASSIGNMENT 2

[Problem Set]

MARASIGAN, VEM AIENSI A
3BSCS-1

1. **Do you believe our capacity for abstract thought is influenced by our language skills? Support your opinion.**
   Yes, language skills significantly influence abstract thinking because having a good foundation of the skill for a particular language enables the capability to understand something, or from a programming language's perspective, it will be easier to decipher the meaning of the program or sequence of codes which will enable to think logically and abstractly, especially regarding a specific situation and problem-solving.

2. **What are some features of specific programming languages you know whose rationales are a mystery to you?**
   So far, since I only programmed in Java, I still don't understand why it has a longer code state, just like when comparing it to Python where "print" is smaller than Java's "System.out.print". I am assuming that it is because Java is multiplatform compatible but still, I'm not sure why the language has that kind of coding way.

3. **What arguments can you make for the idea of a single language for all programming domains?**
   This idea, of having only one language for all programs, has an advantage since if every application uses the same language, communication, or coherence, and might as well compatibility of application will be improved.

4. **What arguments can you make against the idea of a single language for all programming domains?**
   Although, the idea of having only one language for all programs will be less efficient in my view. This is because applications might be unnecessarily larger since they will be using a single language for all. Meaning that it may contain features of the language that are not needed for the application.

5. **Name and explain another criterion by which languages can be judged (in addition to those discussed in this chapter).**
   Compatibility is another criterion to look up to because it is better if a particular programming language can perform multiplatform.

6. **What common programming language statement, in your opinion, is most detrimental to readability?**
   As of now, considering my initial knowledge in programming, it is hard for me to read when incorporating machine-like language in statements such as Ascii or when using decimal, hexadecimal, and binary, as this sometimes makes me confusion making it harder to read or decipher.

7. **Java uses a right brace to mark the end of all compound statements. What are the arguments for and against this design?**
   - For me, in the positive view, this way of ending lines of code with braces is practical because it lets me easily determine the block of code or where those groups of codes belong. Like in loops, conditional statements, and even in methods and object-oriented programming. It also helps define the hierarchy on which statements are included in a particular block and which are involved and not.

- However, looking at the negative perspective, I have not found anything irrelevant or against this design yet during my journey in programming. Maybe in a visual aspect where it looks like parenthesis, which is completely used for different things, but still it is not that of a big issue.

8. **Many languages distinguish between uppercase and lowercase letters in user-defined names. What are the pros and cons of this design decision?**
    - In my perspective in this design, the strict use of Upper and lower cases, is good in terms of uniqueness and definition of data holders. Means that naming conventions can be more efficient, just like; a String word is different from class Word. By this, naming can be repeated rather than thinking for a completely different name for another variable.
    - But still, it has cons too, knowing that naming conventions can be repeated just by using a different case of the word or letter, this may lead sometimes to confusion when not carefully understood. One scenario is when storing data in the String word variable, the programmer typed Word instead of using the word variable. The error may take some time to find the since this naming repetition also decreases readability of the code.

9. **Explain the different aspects of the cost of a programming language.**
    - First is the cost of training programmers, which pertains to the time spent to learn the language considering the language's simplicity as well as the programmers experience.
    - Second is the cost of writing programs, which is also affected by the writability criteria of the language. Meaning that the higher the writability of a program, the lesser the cost of software production.
    - Third is the cost of compiling programs in the language, which I think is also relevant to the cost of executing the written program since a program must be able to be properly compiled by a compiler to be able to run.
    - Fourth is the execution of the programs written in the language which involves the run-time, how fast the program was executed or how long it took to run. Just as cost of compiling programs, this cost also false under the concept of Optimization.
    - The fifth is the cost of implementation whereas a programming language has the capability to be implemented in different platforms to consider it cost efficient.
    - Sixth is the cost of poor reliability which is also another cost to be taken to account as this is significantly important as languages must be reliable or dependable, capable of withstanding operations and algorithms as failure can result to a very-high cost.
    - Lastly is the cost of maintaining programs, which can also fall under the writability criterion as programs requires updates and maintenance over-time. A language must have this capability to efficiently make changes to the program rather than starting again from scratch, which unnecessarily results in a greater cost.

10. **What are the arguments for writing efficient programs even though hardware is relatively inexpensive?**

    Writing efficient programs promotes faster execution and reduces irrelevance. By that, it reduces system resources too providing more resources for other programs to fit. Also, hardware-wise, efficient programs may still run even in earlier hardware's resources.