**CS252 Reading Assignment #1**
**Summary**

The philosophy of author is that the primary purpose of a programming language is to help a programmer in the practice of his art, namely, program design, documentation and debugging. There are five main criteria - simplicity, security (or reliability), fast translation, efficiency of object code and readability. A high level language design should aim at the simplicity and clear modular description of the best hardware designs. He provides the do's and don'ts of designing a programming language:

1) **Comments** - Comments are necessary, but they should not interfere with the actual code of the program.
2) **Syntax** - Syntactic framework chosen should enable faster compilation and make error detection & recovery easier, but also must not increase the complexity of the program.
3) **Arithmetic Expressions -** Permitting the use of expressions with complex structures as operands and results, is not recommended as it adds penalty in efficiency and programmer control. The programmer should be provided with tools to design and implement his own representation of data and code the operations upon it.
4) **Program Structures** - These structures should enable the programmer to update his environment. But, these should not add any extra complexity, or add any scope for tricky programming and even trickier errors.
5) **Variable** - Splitting a machine into number of separate variables, arrays, files, etc. makes it easier to identify part of the machine subject to change in an update operation. The introduction of reference or pointers is a step backward, as it reduces the clear separation between an address and its contents.
6) **Block structure** - There is a close correspondence between the statically visible scope of a variable in a source program and the dynamic lifetime of its storage when the program is run. The introduction of references which may point to variables of an exited block breaks this correspondence.
7) **Procedures and Parameters** - These tools make the language extensible and provide great economies of storage at run time. However, its effect should be clear from the syntactic form and its correctness must be checked rigorously at compile time.
8) **Types** - Automatic type transfers and coercions are complicated, inefficient and leads to unexpected errors. By declaring the type of each variable, any mistake or mismatch can be detected at compile time.

The author believes that designing of a feature and designing of a language should be treated as two separate activities. In the former, the task should be innovation of a new feature, and its rigorous testing. While the latter should deal with consolidation of these features to create a consistent language, which he can sell & distribute to intended audiences.

**Ujjawal Garg** **011917334**