Procedural language

- The program code is written in the form of a sequence of instructions.
- The user would specify what has to be done and how it can be done, i.e the step by step procedure of it.
- It is considered as a command-driven language.
- It works with the state of machine.
- Its semantics are tough in comparison to other paradigms.
- The size of the program would be large.
- These steps would be executed in a sequential method.
- It returns restricted data types and certain allowed values only.
- The overall efficiency is high.
- The instructions are written to solve a specific/set of problems.
- Examples of procedural languages include BASIC, FORTRAN, ALGOL, C, COBOL, and Pascal.
- It is not suited for applications where time is a critical constraint.
- The iterative loops and recursive calls are used while working in procedural languages.

Non-procedural Language

- The user would specify what has to be done but doesn't get into the how it has to be done part.
- It is known as an applicative or functional language.
- It involves developing function based on other functions, in order to construct other complicated functionalities.
- It works with the help of mathematical functions.
- Its semantics are simple in comparison to procedural languages.
- Examples of non-procedural languages include LISP, SQL, PROLOG.
- It is considered as a function-driven language
- It has the ability to return any datatype or value.
- The overall efficiency of non-procedural language is low in comparison to procedural language.
- The programs are small in size.
- It is well-suited for applications where time is a critical factor.
- Recursive calls are used while working with non-procedural languages.