1sub.md

CS3040 Compiler Design

Assignment 01: 26Jan23

1.

Language	*Type*
`C`	imperative, von Neumann, third-generation
`C++`	imperative, object-oriented, von Neumann, third-generation
`Cobol`	imperative, third-generation, object-oriented
`Fortran`	imperative, von Neumann, third-generation
`Java`	imperative, von Neumann, object-oriented, third-generation
`Lisp`	functional, declarative, fourth-generation
`OCaml`	imperative, object-oriented, functional, fourth-generation
`Perl`	imperative, third-generation, object-oriented, scripting
`Python`	imperative, declarative, object-oriented, functional, third-generation, scripting
`VB`	imperative, object-oriented, third-generation

2.

Compiler	Interpreter
Converts source code written in one programming language into another language, usually machine code	Executes source code line by line
The compiled code is in a format that can be run directly on the computer's hardware	The code is executed on the fly and requires an interpreter to be installed
Reads the entire program and searches multiple times for a time-saving execution.	No rigorous optimization takes place because the code is executed line by line.
Generally faster execution time	Generally slower execution time
All the (compile time) errors are shown at the end of the compilation and the program cannot be run until those are resolved.	Displays the errors from line to line. The program runs till the error is found and proceeds further only on resolving.
An intermediate step is required before the code can be run	No intermediate step is required
Examples: `C`, `C++`, `Fortran`	Examples: `MATLAB`, `Ruby`, `Python`

Some languages like `Python` and `Ruby` have both a compiler and an interpreter, which means that the source code can be compiled to bytecode and then run by an interpreter, or it can be interpreted directly.

-o0o0o0o-