PROGRAMMING LANGUAGES

Programming is the task of creating or modifying code that guides the operation of the computer. This is different from installing a program. Installing a programming is loading a program that has already been prepared onto the computer for future use.

Types of Programming Languages

- **Low Level Languages**

These programs are machine dependent. The code is difficult to understand. Examples: machine language and assembly language.

- **First Generation Language**

Machine Language: A low level language recognized and executed by the computer’s CPU. This language consists of 1s and 0s. Machine language programs are difficult to write or modify. However, these programs execute faster and require fewer instructions than high level language programs. The machine code written on one type of computer may not work on a different type of machine.

- **Second Generation Language**

Assembly Language: A symbolic language in which each programming statement corresponds to a command that the CPU can execute. (E.g. MUL may represent multiply). An assembler translates assembly language programs to machine language. Assembly language programs can be easily converted to machine code by an assembler. However, they are difficult to understand when compared with high level languages programs.

❖ High Level Languages

These programs are machine independent. The code uses English-like statements which makes it easier to understand. Hence, high level language programs are easier to create, read and develop than low level programs.

❖ Third Generation Languages (3Gls)

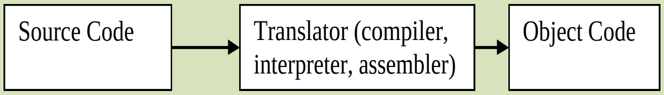
Translators are used to convert these languages to machine code

Examples of Translators:

Compiler: A program which translates high level language programs (source code) to a machine readable executable program (object code). Compiled programs execute faster than interpreted ones because they do not have to share memory space with the interpreter. **COBOL, Pascal , Fortran, Visual Basic and C are programming languages that use a compiler**.

Interpreter: A program which does not translate source code in a single pass but rather translates and executes each source code instruction before translating and executing the next. An interpreter does not create object code. Interpreted programs run slower than compiled ones because the interpreter must be present for the code to run. Since an interpreter reads and executes code, it is faster than a compiler when it is being used to test and develop programs. **Basic is an example of a programming language that uses an interpreter.**

Diagram to showing source code being translated to object code.



❖ Fourth Generation Languages (4GLs)

These are non-procedural languages that allow users to design reports, screens and databases. 4GL use English-like statements which make programs easy to write. Consequently, programs can be written quickly and software development cost is reduced.

Example	**Use**
Structured Query Language (SQL)	Database
Visual Basic	Graphical User Interface (GUI)
COBOL	Business

Example of SQL programming: Use StockTable

Select All Items Where Price > $10.00

❖ Fifth Generation Languages (5GLs)

These languages provide graphical tools for the creation of bodies of programming code. 5GLs are non-procedural. The programmer states the goal to achieve but not the steps required to achieve the goal. 5GLs are designed to make the computer solve the problem for you. **Examples: Prolog (Logical Programming, OPS5 (Official Production System) and Mercury.**

❖ Natural Language Programming

Programs are written using expressions similar to your native language. Such languages will require a large number of command words to accommodate the wide range of expressions in the human language. This system will also have to learn new expressions. Natural language processing involves a few expressions as opposed to natural language programming which uses several expressions.

Programming Languages			
Language	**Use**	**Language**	**Use**
COBOL	Business Applications	FORTRAN	Scientific Applications
BASIC	Teaching	Logo	Teaching Children
JAVA	Internet Search Software	Pascal	Teaching
HTML	Develop Web sites	PROLOG	Artificial Intelligence
C	Operating Systems	C++	Operating Systems