Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X
IMPACT FACTOR: 7.056

IJCSMC, Vol. 9, Issue. 9, September 2020, pg.50 – 60

A Short Communication on Computer Programming Languages in Modern Era

Dr. Mamillapally Raghavender Sharma

Assistant Professor, Department of Statistics, University College of Science, Osmania University, Hyderabad, Telangana, India, e-mail: drmrsstatou@gmail.com

DOI: 10.47760/IJCSMC.2020.v09i09.006

Abstract— A Computer Programming Language is an artificial language designed for communicate with a computer in order to control the flow of operations of computer using instructions or commands. In other words we can say a programming language is the process of writing, testing, debugging, and maintaining the source code of computer programs. A computer can be programmed to complete a variety of task using specific language paradigms. A simple programming language facilitates easier communication between the computer and the computer user. So far, in developed programming languages, high level programming languages occupies tremendous place and spread around globe in ubiquitous computing environment. In this paper we present a simple note on programming languages and their translators with the explanation of algorithms and flowcharts.

Keywords- Programming, Syntax, Computation, Code, MU, Multi-tasking, MLL, HLL, Algorithm, Flowchart

I. INTRODUCTION

Generally, to communicate with others and to express our feelings with another we need proper way of communication. This proper way of communication denotes a natural language. Some basic natural languages which used in general communication among humans are English, Hindi, Oriya etc. But using natural languages we cannot communicate with a computer because it cannot understand that natural language for transfer of data or information. So, a proper way of communication is required to communicate with computer to do specific task. This proper way of communication is a vocabulary and set of grammatical rules (syntax) to instruct a computer is termed as a programming language.

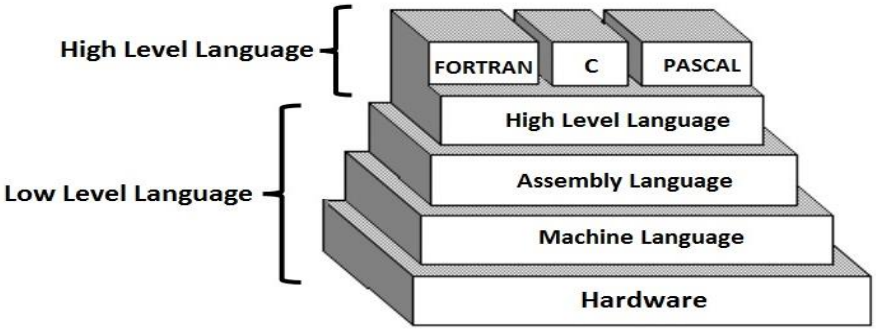


Fig. 1 Computer languages categorization

© 2020, IJCSMC All Rights Reserved

A programming language performs specific task either in simple or complex computations. Instruction or commands vary from prototypes that are used in a program. This programming concept was originated and started ever since the invention of difference engine in 1822 by father of computer who was Charles Babbage. This common reason intended to the creation of many programming languages. All types of specifications will be stored in the computer MU (Memory Unit) in the form of a program. A program is a sequence of specific instructions or specific commands which specify a computation provided by a programmer to control the operation of a computer following the rules of a chosen language i.e. a program contains a list of variable and a list of statements that tells the computer what to do and how to do a specific task. Moreover, a computer program tells the computer how to interact with the programmer (user), hardware, and process the information. Without programs computers are useless. Writing instructions in a program are written in a particular way depend on the type of task. For example, FORTRAN and C languages are used for scientific applications, whereas COBOL is used for business data processing. A programming language allows a programmer to instruct a computer with commands (codes) that both computer and the programmer can understand. A programming language whose data types numbers and arrays may be natural for determine numerical issues. All available data tools can be categorized as:

- Programming languages,
- Mathematical analysis, and
- Visualization tools.

Developing a programming method and analysing methods depend on your background knowledge, but you can simply, easily work with visualization tools without any background knowledge.

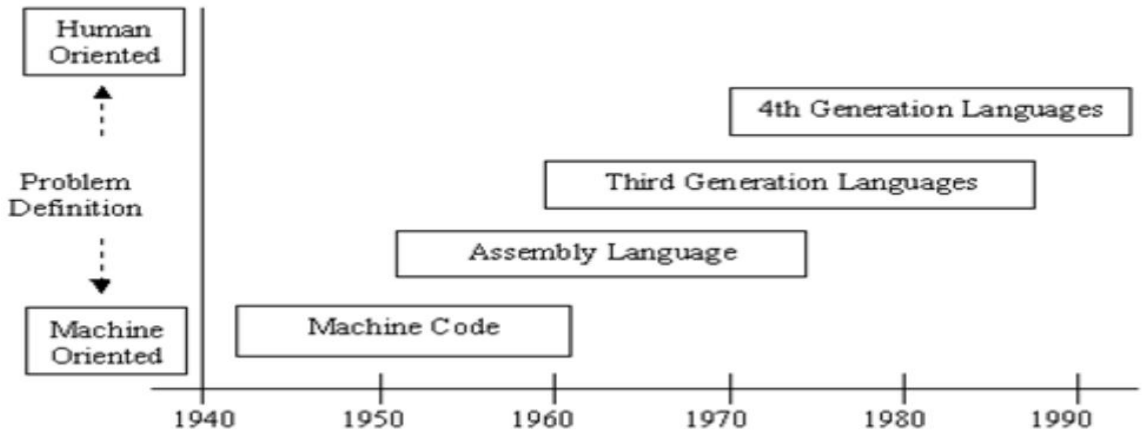


Fig. 2 Relativity of computer languages generations with time scale

The remainder of this paper is organized as follows. Section II describes state of the art. And section III explains methods which explain about the basic idea of existed programming language tools. Section IV summarizes the conclusion. At last we have given references which are used in this paper.

II. STATE OF THE ART

In the history of computer languages, many languages have been created so far and new languages are being created every year. The history of programming languages can be tracked to the development of Charles Babbage's difference engine which was made for execution of specific tasks based on mathematical calculations in the form of physical motion. This form of physical motion has been replaced by electric signals when the U.S Government built the ENIAC in 1942.

Fig.3 Computer assembler working mechanism

A) Computer languages categorization

In a computer program instructions can be executed directly when they are in computer machine language, after a substitution process when in a assembly language, or after translation from some high level language.

© 2020, IJCSMC All Rights Reserved

Although, there are many languages available for our convenient and for better understanding, we categorize some important programming languages as:

1) ***Algorithmic language:*** Algorithmic languages are designed to express mathematical or symbolic computations. They express algebraic operations in notation similar to mathematics and allow the use of subprograms. They were the first high-level programming languages.
   Example: FORTRAN, ALGOL, LISP, C

2) ***Business oriented languages:*** These languages are used for business and data processing purpose.
   Example: COBOL, SQL,

3) ***Educational-oriented languages:*** These languages are used for Educational, and teaching purpose.
   Example: BASIC, Pascal, LOGO, Hypertalk

4) ***Object-oriented languages:*** Object Oriented languages help to manage complexity in large programs. Objects package data and the operations on them so that only the operations are publicly accessible and internal details of the data structure are hidden. In addition, objects may be derived from more general ones, inheriting their capabilities.
   Examples: C++, Ada, Java, Visual Basic etc.

5) ***Declarative languages:*** declarative languages also called as non-procedural or very high level languages in which a program specifies what is to be done rather than how to do it. In such languages there is less difference between the specification of a program and its implementation than in the procedural languages described so far. The two common kinds of declarative languages are: logic and functional languages.
   Examples: PROLOG (Programming in Logic), SQL database language, LISP, and Haskell

6) ***Scripting languages:*** Scripting languages are called as little languages. They are intended to solve relatively small programming problems that do not require the overhead of data declarations and other features needed to make large programs manageable. These are used for OS utilities.
   Example: PERL (practical Extraction and Report Language)

7) ***Document editing languages:*** These languages are specifying the organization of printed text and graphics. They fall into many classes: text formatting notations, description languages, and Mark-up languages.
   Examples: TeX, PostScript, SGML

8) ***World Wide Web Display Languages:*** These languages are used to work with World Wide Web.
   Example: HTML, XML

9) ***Web Scripting Languages:*** Web pages marked-up with HTML or XML are largely static documents. Web scripting can add information to a page as a reader uses it or let the reader enter information that may, for example, be passed on to the department of an online business. CGI (Common Gateway Interface) provides one mechanism: it transmits requests and responses between reader's web browser and the Web server that provides the page. Another approach is to use a language designed for web scripts to be executed by web browser.
   Examples: Javascript, VB script

TABLE I
COMPARISON BETWEEN LOW AND HIGH LEVEL LANGUAGES

	LLL		**HLL**
	Machine Language	**Assembly Language**	**High level language**
Execution	Direct	Using Assembler	Using a Compiler or an Interpreter
Process Time	Takes more time	Simpler language than ML	Simpler than ML and AL

In 1954 IBM established a project directed by J. W. Backus to develop a computer compiler. The project resulted in the creation of FORTRAN language in 1957 which was intended for scientific computations with real numbers and collection of them organized as one-or multi-dimensional arrays . This language had a notion of orientation to the mathematicians and scientists. Although, it was good at handling of numbers, it was not so

© 2020, IJCSMC All Rights Reserved

good at handling of I/O operations. In the year 1958 John McCarthy of MIT (Massachusetts Institute of Technology) created the LISP (List Processing) language for the research purpose of AI (artificial intelligence). A LISP program is a function applied to data, rather than being a sequence of procedural steps as in FORTRAN and ALGOL. After one year in 1959, a new language has been developed for data processing named COBOL (Common Business oriented language). COBOL uses English like notations when introduced. Business computations organize and manipulate large quantities of data, and COBOL introduced the "record data structure" for such tasks. Records are an important example of chunking data into a single object and they appear nearly in all modern languages. In 1960, ALGOL (Algorithmic language) language was created by American and European computer scientists committee for scientific use by introducing block structure in which programs are composed of blocks that might contain both data and instructions. Block structure became powerful tool for building large programs out of small components. Although, it was good at formal grammar, it is difficult to use for general purpose programming. Due to this reason, many languages adopted and developed such as C, C++, and Java. Dr. John G. Kemeny and Thomas Kurtz developed the BASIC (Beginner's all purpose Symbolic Instruction Code) language at Dartmouth in 1964 with the objective to create a simplified computer language for teaching students how to program with computer. It had simple data structures and notations and it was interpreted. Its small size and simplicity made BASIC as a popular language that time personal computers.

In the year 1966, IBM (International Business machine) developed a common programming language with multi-tasking feature to meet the requirements of both scientific, commercial users, known as PL/I (Programming Language/I). Then after, the development of PASCAL language by Niklaus Wirth in 1968 was mainly out of necessity for a good teaching tool for debugging and editing system and support for common early microprocessor machines which were using in teaching institutions. This language was designed in a very orderly approach, by combining many of best features of the languages used in that time like COBOL, FORTRAN, and ALGOL. Dennis M. Ritchie created the C language at AT&T Bell Labs in 1972. This language includes all the features of generality. Actually, Ritchie developed C for new UNIX system being created at the same time. But it became the most common general purpose language to program all OS such as UNIX, Windows, the MacOS, and Linux. In 1980, a new programming method known as OOP (Object Oriented Programming) with an idea of representing pieces of data in the form of objects that can be packaged and manipulated by the programmer. Based on this idea Bjarne Strousturp in 1983 extended the features of C language to develop a full featured language known as C++. Microsoft released a graphical version of BASIC language that simplifies the writing of program for Windows and named it as VB (Visual Basic). In 1995, the Java language hit the scene rapidly rose the popularity and is widely used truly object oriented programming language in existing today. In 2000, the expansion of WWW (World Wide Web) leads to the demand of internet programming. As a result many programming languages such as Perl, PHP, and Python were developed and used by programmers to develop the web applications.

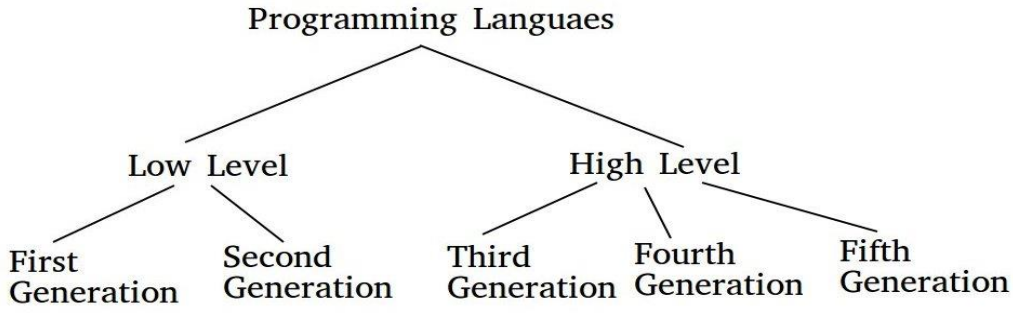


Fig.4 Computer languages grouping chart

SQL (Structured Query Language) is a language for specifying the organization of database. A database is a collection of records (database). Databases organized with SQL are called as relational because SQL provides the ability to query a database for information that falls in a given relation. Commercial database programs commonly use SQL-like languages for queries. LOGO originated in the late 1960s as a simplified LISP dialect for education purpose. Seymour Papert and others used it as the teaching aid to teach mathematical thinking to school children at MIT. It had more conventional syntax than LISP and featured turtle graphics, a simple method to generate computer graphics. Together with recursive routines, it was made a simple computer graphics generator to program intricate and attractive patterns. Hypertalk was designed as "programming for rest of us" by Bill Atkinson for apple's Macintosh. Using a simple English-like syntax, it enables anyone to combine text, graphics, and audio quickly into linked stacks that could be navigated by clicking with a mouse on standard buttons supplied by the program. It was particularly popular among educators in the 1980s and early 1990s for classroom multimedia presentations. ADA, the language was developed in the early 1980s for the U.S

© 2020, IJCSMC All Rights Reserved

Department of Defence for large-scale programming. It combines Pascal-like notations with the ability to package operations and data into independent modules. Its first form Ada-83 was not fully object-oriented but subsequent Ada-95 provided objects and the ability to construct hierarchies of them. In the early 1990s, Java was designed by Sun Microsystems, Inc., as a programming language for WWW (World Wide Web). Although it resembles C++ in appearance, it was fully object-oriented. In particular, Java dispensed with lower-level features, including the ability to manipulate data addresses, a capability that is neither desirable nor useful in programs for distributed systems. In order to be portable, Java programs are translated by a JVM (Java Virtual Machine) specific to each computer platform. In addition to adding interactive capabilities to the internet through web applets, it has been widely used for programming small and portable devices, such as mobiles.

Visual Basic was developed by Microsoft to extend the capabilities of BASIC by adding objects and event driven programming (buttons, menus, and other elements of GUIs (Graphical User Interfaces). VB can also be used within other Microsoft software to program small routines. PERL (Practical Extraction and Report Language) was developed in the late 1980s; originally use with the UNIX OS. It was intended to have all the capabilities of earlier scripting languages. It provides many ways to state common operations and thereby allowed a programmer to adopt any convenient style. It was the most popular system-programming tool during 1990s.

TeX was developed during 1977-86 as a text formatting language by Donald Knuth, a Stanford University professor to improve the qualities of mathematical notations in his books. It replaced earlier text formatting languages due to its powerful and flexible abilities. PostScript is a page-description language developed in early 1980s by Adobe systems Incorporated on the basis of work at Xerox PARC (Palo Alto Research Centre). It uses postfix, also called reverse polish notation in which an operation name follows its arguments. The success of postscript is due to its specification's being in the public domain and to its being a good match for high-resolution laser printers. It has influenced the development of printing fonts, and manufacturers produce a large variety of postscript fonts. SGML (Standard Generalized Mark up Language) is an international standard for the definition of mark-up languages; that it is a Meta language. Mark-up consists of notations called tags that specify the function of a piece of text or how it is to be displayed. It emphasizes descriptive mark-up, in which a tag might be <emphasis>.it is used to specify DTDs (Document Type Definitions). A DTD defines a kind of document, such as a report by specifying what elements must appear in the document. A marked-up text may be analysed by a parsing program to determine if it conforms to a DTD.

The WWW is a system for displaying text, graphics, and audio retrieved over the internet on a computer monitor. Each retrieval unit is known as a Web page. HTML (Hyper Text Mark-up Language) is the mark-up language for encoding web pages. It was designed by Tim-Berners-Lee at the CERN nuclear physics laboratory in Switzerland during the 1980s and is defined by an SGML DTD. HTML mark-up tags specify document elements such as headings, paragraphs, and tables. They mark-up a document for display by a computer program known as a web browser. The browser interprets the tags, displaying the headings, paragraphs, and tables in a layout that is adapted to the screen size and fonts available to it. HTML does not allow one to define new text elements; that is, it is not extensible. XML (Extensible Mark-up Language) is a simplified form of SGML intended for documents that are published on the web. Like SGML, XML uses DTDs to define document types and the meanings of tags used in them. XML provides more kinds of hypertext links than HTML, such as bidirectional links and links relative to a document subsection.

B) Generations of programming languages

The concept of language generations is closely related to the advances in technology. Programming language is the primary tool for creating software. Programming languages have been developed over the years in a phased manner. Each phase had made the programming language more user-friendly, easier to use, and more powerful. Each phase of improved made in the development of the programming languages can be referred as a generation. As of 2002, hundred of languages exist, some used than other, and each claiming to the best. All generations of programming languages so far developed are categorized as:

- Machine language
- Assembly language
- High-level language and
- Very high level-language

© 2020, IJCSMC All Rights Reserved

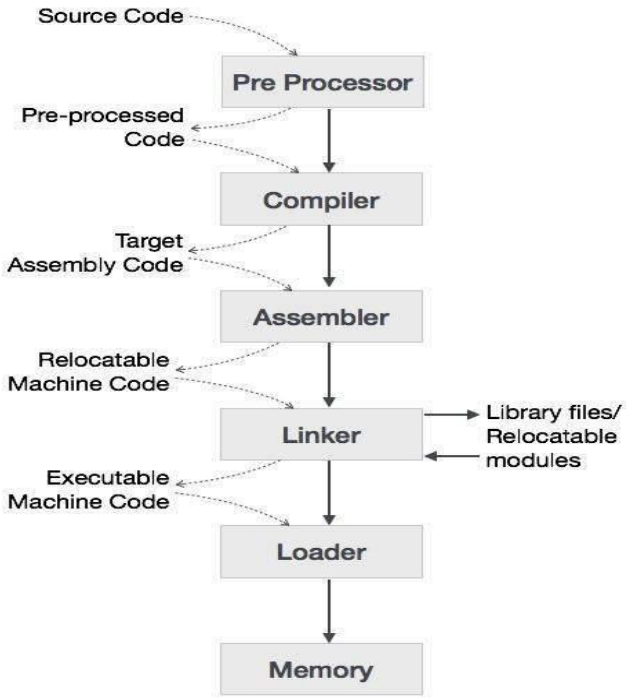


Fig. 5 Computer language processing system

The programming language in terms of their performance reliability and robustness can be categorized into five generations and these generations grouped as:

Low Level Language

- 1GL (Machine Language)
- 2GL (Assembly Language)

High Level Language

- 3GL (Procedural-Oriented Language)
- 4GL (Problem-Oriented Language)
- 5GL (Natural Language)

1) ***1GL (1 Generation Language):*** The first generation languages are low-level languages that are machine dependent languages. They were used to the computer system at a very low level of abstraction. Programming of the first stored program computer systems was performed in machine language. The machine language is referred as native language of the computer system and is referred as the first generation language paradigm. In machine language, all instructions, memory locations, and characters are represented in strings of zeros and ones. Although machine language programs are typically displayed with the binary numbers translated into the octal (base-8) and hexadecimal (base-16), these programs are not easy for humans to read, write, debug, and understand.

   Advantages of 1GL

   - They are translation free and can be directly executed by the computers.
   - The Programs written in these languages are executed very speedily and efficiently by the CPU of the computer system.
   - The programs written in these languages utilize the memory in an efficient manner because it is possible to keep track of each bit of data.

2) ***2GL (2 Generation Language) :*** Assembly language was developed in the mid-1950s and was considered a great leap forward because it uses mnemonic codes or easy to remember abbreviations rather than numbers. The programming process became easier with the development of assembly language, a language that is logically equivalent to machine language but is easier for people to read, write, debug, and understand. The second generation programming language also belongs to the category of low level programming language. The second generation language comprises of assembly language that use the concept of mnemonics for writing the program. Assembly languages are symbolic programming languages

© 2020, IJCSMC All Rights Reserved

that use symbolic notation to represent the machine language instructions. The format of the statement and the exact instructions available will vary from machine to machine because the language is directly related to the internal architecture of the computer and is not designed to be machine dependent. Normally, an assembly language statement consists of a label, an operation code, and an operand. Labels are used to identify or reference the instructions in a program. The operation code is a symbolic notation that specifies the particular operation to be performed such as move, add, subtract, and compare. The operand represents the register or the location in the memory where the data to be processed.

Advantages of 2GL

- It is easy to develop understand and modify the program developed in these languages are compared to those developed in the first generation languages.
- The programs written in these languages are less prone to errors and therefore can be maintained with a great ease.
- The principal advantage of assembly language is that programs can be very efficient in terms of execution time and main memory usage.

3) ***3GL (3 Generation Languages):*** The third generation programming languages were designed to overcome the various limitations of the first and second generation programming languages. The languages of the third and later generations are considered as high-level language such as C because they enable the programmer to concentrate only on the logic of the program without considering the internal architecture of the computer system. A programming language in which the statements are not closely related to the internal characteristics of the computer is called as high-level language. Third generation languages can be platform independent i.e. the code written for one system will work on another system. To convert a third generation program object code requires a compiler or an interpreter. Some of third generation languages are imperative languages means in which language code is executed by line by line in a programmer defined order. Procedural programming is a methodology for modelling the problem being solved, by determining the steps and the order of those steps that must be followed in order to reach a desired outcome. These languages are designed to express the logic.

Advantages of 3GL

- It is easy to develop, learn, and understand the program.
- As the program written in these language are less prone to errors and thus they are easy to maintain.
- The programs written in these languages can be developed in very less time as compared to the first and second generation language.

Examples: FORTRAN, Algol, COBOL, C, C++, C#, Java, VB, Delphi

4) ***4GL (4 Generation Languages) :*** The fourth generation languages were designed and developed to reduce the time, cost, and effort needed to develop different types of software applications. The 4GL are languages that consist of statements similar to statements in a human language. The languages of this generation were considered as very high-level programming languages require a lot of time and effort that affected the productivity of a programmer. These languages are commonly used in database programming and scripts. It allows the programmer to specify what the output should be, without describing all the details. Five basic types of tools fall under the 4GL category.

- Query languages
- Report generators
- Application generators
- Decision support systems and financial planning languages
- Some micro-computer applications such as Lotus 1-2-3, dbase IV, and Symphony.

Advantages of 4GL

- These programming languages allow the efficient use of data by implementing the various databases.
- They require less time, cost, and effort to develop different types of software applications.
- The programs developed in these languages are highly portable as compared to the programs developed in the languages of other generations.

Examples: Perl, PHP, Python, Ruby, CSS, and SQL, Mat Lab, R, Data Flex

© 2020, IJCSMC All Rights Reserved

TABLE II
4GL AND THEIR CODES WITH GENERALITY

Generation	First	Second	Third	fourth
code example	**10101010011000101 10011010100000010 11111111101000101**	**LDA 34 ADD #1 STO 34**	**X = X + 1**	**body. top { color : green; font-style : bold }**
Language	**Low(Machine code)**	**Low (Assembly code)**	**High (Pascal, C etc.)**	**Very high (SQL, CSS etc.)**
Generality		**One to one**	**One to many**	**One to many**

5) ***5GL (5 Generation Languages):*** The 5GL are designed to make the computer solve a given problem without the programmer. The text of a natural language statement very closely resembles human speech. The programming languages of this generation mainly focus on constraint programming means it works based on problem solving using constraints given to the program, rather than using an algorithm. The 5GL language is a programming language that use visual or graphical development interface to create source language that is usually compiled with 3GL or 4GL language compiler to develop a program. The major fields in which the 5GL are employed in AI research, and ANN (Artificial Neural Networks)

Advantages of 5GL

- These languages can be used to query the database in a fast and efficient manner.
- In this generation languages, the user can communicate with system in a simple and easy way.
- These are non-procedural, so users concentrate on what instead of how.

Example: Prolog, and OPS5

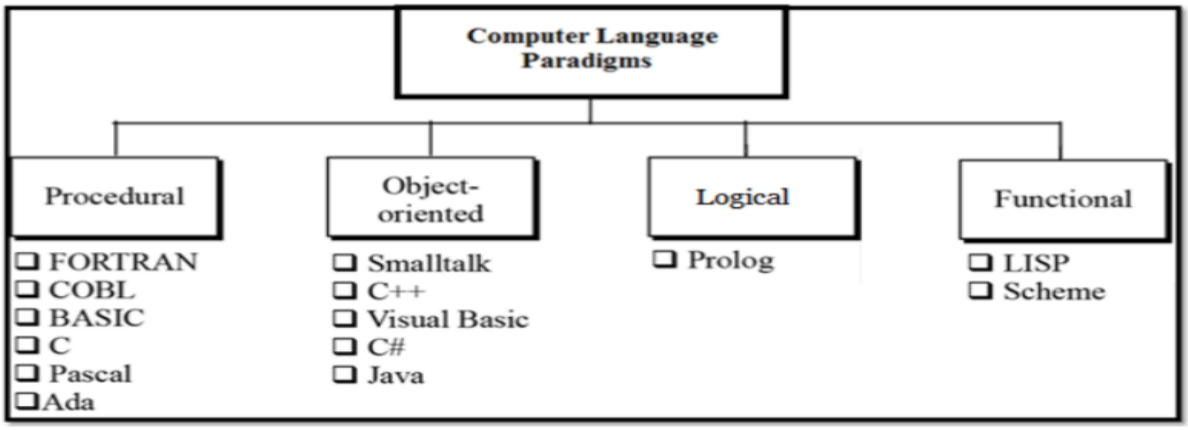


Fig. 6 Simple Anatomy of computer programming languages paradigms

III. METHODS

People express themselves using a language that has many words. But computers understand only a simple language (Machine Language) that consists of only 1s and 0s, with a meaning 1 for "on", and 0 meaning "off". This is because electrical devices seem to fall naturally into one of two possible states.

As the computer operates using a program coded in 0s and 1s, it would be very time consuming process for a human to write program. If the program can be written in a language approaching in an English language or in other or combination of both then the programmer can concentrate more on programming logic and less on programming details. Thus the program should be less likely to contain errors. The effort to overcome this difficulty has lead to the development of new programming languages that can be well suited to the capabilities of both. Languages that favour humans are terms as high-level languages, and those oriented to machine are low level languages.

A) Programming languages categories

There are two major types of programming languages to complete specific required tasks.

- Low Level Languages (LLL): LLL are also called As MLL (Machine Level Languages) that are understood only special numbering systems to execute programs. These are machine oriented and

© 2020, IJCSMC All Rights Reserved

require extensive knowledge of hardware and its configuration. These LLL use binary digits to execute programs. It is very difficult to write a program in Machine code correctly. So, to overcome this big disadvantage assembly language came into existence. But this also slow in program executions.

- High Level Languages (HLL): to overcome the demerits which are in LLL, high level languages introduced into world. By these languages, programs are written in easy way in humans speaking languages. These languages convert source code by using special softwares called translators.

B) LLL categories

Low Level Language furthered divided into:

- Machine Language and
- Assembly Language

Computer programs are written in the earlier days using Machine and Assembly Languages. But it is difficult to write programming in low level languages. This problem overcome when high level languages came into existence. A high level language is easily understandable by humans, and understandable by a computer using translators of that programming language.

C) HLL categories

Based on programming paradigm high level languages are categorized into four important categories as:

1. Imperative languages
2. Procedural languages
3. Declarative languages
4. Object-oriented languages
5. Logical languages
6. Functional languages

D) HLL Translators

To convert a program code into equivalent machine code high level languages use some translators such as:

- Compiler
- Interpreter

E) Algorithms & Flowcharts

To cope with a problem we need to make a better plan which will be convenient and easy way to solve that particular problem. Algorithms and flowchart are two basic terms which aids for the development of a software package conveniently. An algorithm is a step wise set of finite instructions written to solve a problem and a flowchart is graphical representation of an algorithm.

F) Flowchart symbols

Flowcharts are usually drawn using some standard symbols; however, some special symbols can also be developed when required. Some standard symbols, which are frequently required for flowcharting many computer programs are shown.

Terminator: An oval flow chart shape indicates the start or end of the process, usually containing the word "Start" or "End".

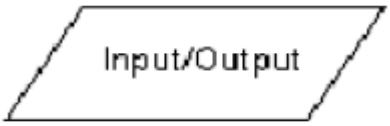
Process: A rectangular flow chart shape indicates a normal/generic process flow step. For example, "Add 1 to X", "M = M*F" or similar.

© 2020, IJCSMC All Rights Reserved

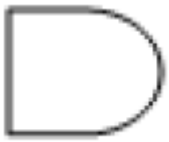
Decision: A diamond flow chart shape indicates a branch in the process flow. This symbol is used when a decision needs to be made, commonly a Yes/No question or True/False test.

Connector: A small, labelled, circular flow chart shape used to indicate a jump in the process flow. Connectors are generally used in complex or multi-sheet diagrams.

Data: A parallelogram that indicates data input or output (I/O) for a process. Examples: Get X from the user, Display X.



Delay: used to indicate a delay or wait in the process for input from some other process.



Arrow: used to show the flow of control in a process. An arrow coming from one symbol and ending at another symbol represents that control passes to the symbol the arrow points to.

These are the basic symbols used generally. Now, the ***basic guidelines*** for drawing a flowchart with the above symbols are that: In drawing a proper flowchart, all necessary requirements should be listed out in logical order. The flowchart should be neat, clear and easy to follow. There should not be any room for ambiguity in understanding the flowchart. The flowchart is to be read left to right or top to bottom. A process symbol can have only one flow line coming out of it. For a decision symbol, only one flow line can enter it, but multiple lines can leave it to denote possible answers. The terminal symbols can only have one flow line in conjunction with them.

IV. CONCLUSIONS

Programming is a part of software engineering used for producing a program and a list of instructions for the computer system. Hardware and software complexity is making programming harder and harder to reason about the environment in which code is written frustrating the objective of reliable, secure, and maintainable software. So, now systems programmers need to be embracing high level languages which give convenience in creating programs in well defined manner. Based on consideration such as what purpose the program is designed to serve and what languages are already being used in the organizations, the programmer selects programming languages for development of special software. As new languages emerge, we hope the designers will carefully consider the possibility of supporting low level programming and that they might find our work useful. At present, many programming languages evolution continues in both industry and research based on the requirements of current web technologies including mobile applications. Current focus in programming language is increasing support for functional programming, distributed programming, and mechanism for adding security for 5GL.

© 2020, IJCSMC All Rights Reserved

REFERENCES

[1] S. M. Metev and V. P. Veiko, *Laser Assisted Microtechnology*, 2nd ed., R. M. Osgood, Jr., Ed. Berlin, Germany: Springer-Verlag, 1998.

[2] J. Breckling, Ed., *The Analysis of Directional Time Series: Applications to Wind Speed and Direction*, ser. Lecture Notes in Statistics. Berlin, Germany: Springer, 1989, vol. 61.

[3] S. Zhang, C. Zhu, J. K. O. Sin, and P. K. T. Mok, "A novel ultrathin elevated channel low-temperature poly-Si TFT," *IEEE Electron Device Lett.*, vol. 20, pp. 569–571, Nov. 1999.

[4] http://www.sanjayutility.com/Technology_6GenerationsofProgrammingLanguages.aspx

[5] M. Wegmuller, J. P. von der Weid, P. Oberson, and N. Gisin, "High resolution fiber distributed measurements with coherent OFDR," in *Proc. ECOC'00*, 2000, paper 11.3.4, p. 109.

[6] Blissmer, Robert H. Introducing Computers: Concepts, Systems, and Applications. John Wiley & Sons, Inc., New York, 1992.

[7] R. E. Sorace, V. S. Reinhardt, and S. A. Vaughn, "High-speed digital-to-RF converter," U.S. Patent 5 668 842, Sept. 16, 1997.

[8] (2002) The IEEE website. [Online]. Available: http://www.ieee.org/

[9] M. Shell. (2002) IEEEtran homepage on CTAN. [Online]. Available: http://www.ctan.org/tex-archive/macros/latex/contrib/supported/IEEEtran/

[10] *FLEXChip Signal Processor (MC68175/D)*, Motorola, 1996.

[11] "PDCA12-70 data sheet," Opto Speed SA, Mezzovico, Switzerland.

[12] A. Karnik, "Performance of TCP congestion control with rate feedback: TCP/ABR and rate adaptive TCP/IP," M. Eng. thesis, Indian Institute of Science, Bangalore, India, Jan. 1999.

[13] https://www.britannica.com/technology/computer-programming-language/Visual-Basic

[14] J. Padhye, V. Firoiu, and D. Towsley, "A stochastic model of TCP Reno congestion avoidance and control," Univ. of Massachusetts, Amherst, MA, CMPSCI Tech. Rep. 99-02, 1999.

[15] *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification*, IEEE Std. 802.11, 1997.

[16] Brightman, Richard W., and Jeffrey M. Dimsdale. Using Computer is an Information Age. Delmar Publishers Inc., NY, 1986.

[17] E. Balagurusamy, Fundamentals of Computers, McGraw Hill Education (India), 2009.

[18] https://en.wikibooks.org/wiki/A-level_Computing/AQA/Paper_2/Fundamentals_of_computer_systems/Classification_of_programming_languages

Authors Profile

Dr. M. Raghavender Sharma (drmrsstatou@gmail.com) completed Bachelor of Science in Mathematics, Master of Science in Statistics, and achieved Doctoral Degree in Statistics, all degrees from Osmania University, Hyderabad, Telangana, India. Currently he is working as an assistant professor in the department of statistics at University College of Science in Osmania University, Hyderabad, Telangana, India. His main research work focuses on Time Series Analysis (Statistics). He is supervising many PhD Scholars (in both statistics and Computer Science subjects) and he has published 33 articles in national, international peer reviewed journals and participated presented many articles in conferences and seminars. He has excellent teaching track record with 30 years teaching experience and 25 years Research experience.

© 2020, IJCSMC All Rights Reserved