Computer languages:-

Computer language is a set of syntactic and semantic rules with well defined vocabulary used to define computer programs for instructing a computer to perform tasks. A language enables a programmer to precisely specify what data a computer will act upon, hoe there data will be stored or transmitted, and precisely what actions to take under various circumstances.

The term computer language includes a wide variety of languages used to communicate with computers. It is broader than the more commonly used term programming language. Programming language are a subset of computer language. For example, HTML is a markup language and a computer language, but it is not traditionally considered a programming language. Machine code is a computer language that can technically be used for programming (e.g. for mant system utilities) and most woulddnot consider it a programming language.

Computer languages can be divide into two groups : high-level languages and low-level languages. High-level languages are designed to be easier to use, more abstract and more portable than low-level languages. Syntactically correct programs in some languages are then compiled to low-level language and executed by the computer. Most modern software is written in a high-level language, compiled into object code, and then translated into machine instructions.

Machine language:-

A machine language consists of the numeric codes for the operations that a particular computer can execute directly without any translation. Machine language are the only languages understood by computers. The codes are strings of 0s and 1s for this language. For the first machines in the 1940s, programming had no choice but to write in the sequences of digits that the computer executed.

For example, assume we want to compute the value of A + B + C, where A is the value at machine address 5012, B is the value at address 5013, and C is the at address 5015. It should be clear that programming in this manner is difficult and possible with errors. Explicit memory locations must be written, and it is not always obvious if simple errors are present. For example, at location 5012, writing101… instead of 111… would compute |A + B + C| rather than what was desired is not easy to detect. It was very soon replaced by slightly improved assembly language.

Assembly language:-

It uses short mnemonic codes for instructions and allows the programmer to introduce names for blocks of memory that hold data. They associate a symbolic name to the machine language code, for example :ADD BX, 12 ; MOVCX, 25 etc. since each component of a program stands for an object that the programmer understands, using its name rather than numbers should make it easier to program. By naming all locations with easy-to-remember names, and by using symbolic names for machine instructions, some of the difficulties of machine programming can be eliminated.

A relatively simple system program called an assembler converts this symbolic notation into an equivalent machine language program. The symbolic nature of assembly language greatly eased the proframmer’s burden, but program were still very hard to write. Mistakes were still common. Programmers were forced to think in terms of the computer’s architecture rather than in the domain of the problem being solved.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A+  ABC  Ada  Alan  ALF  Algol  Alloy  Amiga E  AMPL  APL  AWK  BEIf  BASIC  BCPL BETA Bliss  Blue  Business Rules !  CFortrees  C++  Cecil  Centum  Charity  CHILL  CLAIRE | Clean  COBOL COMAL Corn  cT  D  DCL Dylan  E  Eiffel  Elastic  Lingo  Erlang  Euphoria  Felix  Ferrite  Forth  Fortan  Mercury  Fril  GNUE  Guile  Godel  Haskell  Hugo | ICI  ICON Inform  J  Java  Joy  K  LabVIEW  Lava  LIFE  Limbo  Phantom (Phi)  LISP  LOGO Lua  Matlab  MC#  MCPL  Postscript  Miranda  ML Module -2  Module -3  Nemerle | NeoBook  NESL  NetRexx  Oberon-2  Objective-C  Oblig  Occam  Octave  Oz  Pascal  Perl  SNOBOL  PHP  Pike (LPC)  PiXCL  PL/b  PL/I  Plaint  Visual Basic  Prolog  Python  Q  R  REBOL  Rexx | RPG  RPL/2  Ruby  S  Sather  Scheme  Self  SETL  Simula  Sisal  Smaltalk  SR  TADS  Tel  Theta  TOM  Turing  WinBatch (WIL)  Yorick  ZPL |

High-level/programming language:-

Hogh-level language is such that which supports the development without worrying about lots of details that are irrelevant to the problem at hand (i.e. details about system architecture). Each programming language can be thought of as a set of formal specifications concerning syntax, vocabulary, and meaning. Program instructions are organized through a unique set of keywords (words that it understands) and a special syntax. Any programming language specifications usually include :

1 Data and data structures

2 instruction and control flow

3 Reference mechanisms and Re-use

4 Design philosophy

Some Imoprtant Programming Languages

1. Autocode – 1952 : Alick E. Glennie.

This is a symbolic code language implemented firstly on mark 1, then on other computers.

2. FORTRAN (FORmula TRANslator system) -1954: 1958 – John Backus and other Researchers at IBM.

This language is dedicated to mathematical calculations. Fortran II (1958) introduced SUBROUTINES, FUNCTIONS, LOOPS, a primitive FOR control structure. Identifirers were limited to six characters.

3. LISP (LISt processing) – 1958-1960 : Mac Carthy.

This is a functional language for list processing. It is purely recursive, and not iterative. There is no difference between code and data.

4. ALGOL (ALGOrithmic language) - 1960 / Algol W - 1966 / Algol 68 : Defined by an international consortium of computer science specialists, coordinated by IFIP.

This was the first universal language to be machine independent. This introduced the use of the BNF ( Backus Naur Form) grammar to create a syntax parser. As well as presents BLOCKS of STATEMENTS and LOCAL VARIABLES are inside a block. Recursivity was implemented but with reticence as this was considered as useless!

Uses DTNAMIC ARRAYS,and this, means that following language(Pascal, C) have regressed by using static arrays, for better performance. It used all the traditional statements like-IF THEN ELSE, FOR, the: =symbol for assignment (used then by pascal), a SWITCH with GOTOs the BEGIN END markers, the WHILE loop etc.

5. COBOL (COMMON BUSINESS ORIENTED LANGUAGE) - 1960 : Defined by a committee, the CODASYL, Conference on Data Systems languages.

Grace Murray Hopper. Who had desired flow-matrix, a compiled language in the 50’s, has participated to the committee for COBOL with department of defence and others in may 1959 to april 1960. COBOL is a classical procedural language aimed at enterprise management, in which a program is divided in 4 division : identification, environment, data procedure, and they may be divided in sections. It was founded on data and a program must describe precisely the hardware and inout/output data format.

It introduced the RECORD data structure. Programs are documented by the syntax and are all but lightweight.

6. Basic (beginner’s all propose symbolic instruction code) - 1964 : John Kemeny, Thomas Kurtz.

This language has been designed in 1963 and implemented in 1964. The language is based on interpreter and each line has a number to allow GOTOs statements to jump to the line. Pascal and C have replaced BASIC in the same decade. Microsoft still uses BASIC and its feature like visual basic for applicant, ASP for the web and applicants are extended with a language (macro) that is BASIC also.

7. LOGO – 1966 : Fuerzeig, Seymour papert and others.

It was the first successful kids programming lamguage aimed to teach programming to children and based on moving a “turtle” on the screen.

8. CPL (combined programming language) – Cambridge and London universities.

This was a combination of Algol 60 and functional language aimed at proof of theorems. It was using polumorphic testing structures. It was a first step tpward the design of the C language.

9. BCPL (BASIC CPL) – 1965 : Martin Richards.

This language is an improved version of CPL and uses all traditional statements like – FOR, LOOP, IF THEN, WHILE, UNTIL, REPEAT, REPEAT WHILE, SWITCH CASE etc. it owned procedures and funtions, unlike C and used block delimiters $ (….. $) which has inspired probably /\* … \*/ in C.

9. PL/1 (programming language number one-originally NPL(NEW PROGRAMMING LANGUAGE) ) – 1965 : IBM.

This was the modular and general-purpose language with uniqueness that keywords are reserved only in the context where they are used as keywords. It is more hardware independent than predecessors. Data types are recongnized: picture, file, etc. it also supports built-in : array, structure allows various conditional loops, storage classes are introduced : automatic, static ( llife of the program), controlled, based. Exceptions are also implemented.

10. Pascal ( Named from the French mathematician Blaise Pascal) – 1970 : Niklaus Wirth.

This language was specially designed to ease the building of compilers and to lead teaching by forcing to a structured programming. UCSD pascal, written by a groups of programmers leaded away by Kenneth Bowles is the first version for microcomputers. Pascal compiles programs in P – codet that is portable and interpreted ( as java later). It included a complete development environment, a principle used successfully further by Turbo Pascal.

11. Smalltalk – 1972 : Alan Kay and the Software concept group.

This is the initial language recongnized as a fully object oriented language and runs always inside a graphical environment, with windows, mouse, etc.

12. C – 1973 : Dennis Ritchie : This was the landmark in development of programming language hiftory. BCPL and B was using integer for pointers,but this was not working on the new computers.

BCPL has no type (as PHP or modern scripting languages!). the declaration into, charb were created in C. the += operator comes from Alogol 68 ( but was written =+). In BCPL. A block of statement was enclosed inside the (\* and\*) symbols as the comment in /\* and \*/ and subexpressions inside ( and ). The C language simplified the writing with the { and } symbols, that does the original idea doesn’t remains. Union and cast come from Algol 68. The ++ was already in theB language and keyword”include” comes from PL/I. the preprocessor was implemented in 1973.

13. Ada – 1980+ : Designed by a committee leaded by Jean Ichbiah, for the U.S. Department of defence.

It is a very tough language and inspired by pascal and Algol W. It introduced GENERIGITY of algorithms and a kind of primitive object orientation, but becames really object oriented later. It added the concepts of PACKAGES that are independent modules.

14. C++ - 1981 : James Gosling and other programmers at sun.

Initially in 1991,the object oriented language was known as OAK but was unsuccessful and in 1994 has been rewritten for internet amd renamed JAVA. In 1995 applets are run on navigators then in January 1996. Javasoft distributes JDK 1.0, the java development kit.

Java us a classical platform independent procedural language, near C++. It compiles in bytecode, interpreted on any computer. It is simpler then C++ : one class by file. Automatic memory management, no pointers, no multiple inheritance nor operator overloading, but integrated multi-tasking. It has only dynamic array unlike C and C++.

16. PHP (PERSONAL HOME PAGES HYPERTEXT PROCESSOR) – 1995 : Rasmus Lerdorf.

It is a multi-platforms scripting language, embedded inside HTML. It is close to C but not tyoed. Variables are prefixed by the $ symbol as the shell of UNIX or as perl. The interpreter parses an html that embeds PHP code and delivers a pure html page.

An esxtended library of functions allows webmasters to build dynamic pages. Microsoft uses as equivalent language under windows, ASP, near basic.

17.JAVA SCRIPT ( FIRSTLY NAMED LIVESCRIPT) – 1995 : Brendan Eich at Netscape.

It is most commonly used scripting language to embed procedural code into web pages. It may be used to other applications,XML based language for example. It shares the syntax of C or JAVA, but with untyped variables. The elements of the web page (window, table, etc) are accessed through the document object model.

18. UML (UNIFIED MODELING LANGUAGE) – 1996 – standard by OMG (object management group): Grady Booch, Jim Rumbaugh and Ivar Jacobson.

UML is the union of three modeling languages designed by the three authors above. The language uses a graphical notation to design software ptojects. A source is a diagram expression objects and their interactions a model is made of views and combination of them describes a complete system. The model is abstract and domain-independent.

19. C# ( C-sharp) – 2000 : Anders Hejisberg / Microsoft.

This is object oriented language very similar to java and the main language of the. NET environment, to program internet based application. It also include – garbage collector, no pointer, interfaces, and multi-tasking. The C# compiler produces intermediate language, the MSIL (Microsoft intermediate language) and uses a multi-languages library, the CLR (common language runtime). The DIT NET is basically a global programming environment which includes many compatible languages like VB.NET, C-sharp.NET etc. the originality of the .NET system is that various language source code may be compiled ro MSCL microsoft intermediate language code (a common code) and share their classes.

Language processors