

High Level Programming

Sumit Kumar Agarwal¹, Prabhat Chandra Gupta²

¹C.C.S.I.T., T.M.U.MORADABAD, INDIA

²ASSISTANT PROFESSOR C.C.S.I.T., T.M.U.MORADABAD, INDIA

¹sumitkumaragarwal74@gmail.com

²prabhatchandranitk@gmail.com

Abstract— developing a language processor was, indeed, a very big deal. Computers were slow, didn't have a lot of memory, the language processors had to be written in low-level assembly languages. It wasn't something someone would do in their rooms as a hobby, to put it mildly.

C has "for" loop and "while" loops; it has "if (...) ..." statements, and "if ... else ..." statements; it has a multi way selection statement (" switch () {case...} "). Once you've learnt any one of these Algol family languages, you've really learnt them all.

The data types are again similar. There are the standard integer and real numbers, and characters.

In computer science, a high-level programming language is a programming language with strong abstraction from the details of the computer. In comparison to low-level programming languages, it may use natural language elements, be easier to use, or may automate (or even hide entirely) significant areas of computing systems (e.g. memory management), making the process of developing a program simpler and more understandable relative to a lower-level language. The amount of abstraction provided defines how "high-level" a programming language is.

Keyword- Loop, Data Types, Keywords, Structure.

I. INTRODUCTION

- High-level languages are relatively easy to learn because the instructions bear a close resemblance to everyday language, and because the programmer does not require a detailed knowledge of the internal workings of the computer.
- Each instruction in a high-level language is equivalent to several machine-code instructions, therefore it is more compact than equivalent low-level programs.
- High-level languages are used to solve problems and are often described as problem-oriented languages

EXAMPLES OF HLL-

- BASIC WAS DESIGNED TO BE EASILY LEARNT BY FIRST-TIME PROGRAMMERS;
- COBOL IS USED TO WRITE PROGRAMS SOLVING BUSINESS PROBLEMS;
- FORTRAN IS USED FOR PROGRAMS SOLVING SCIENTIFIC AND MATHEMATICAL PROBLEMS.

- WITH THE INCREASING POPULARITY OF WINDOWS-BASED SYSTEMS, THE NEXT GENERATION OF PROGRAMMING LANGUAGES WAS DESIGNED TO FACILITATE THE DEVELOPMENT OF GUI INTERFACES;
- FOR EXAMPLE, VISUAL BASIC WRAPS THE BASIC LANGUAGE IN A GRAPHICAL PROGRAMMING ENVIRONMENT.
- SUPPORT FOR OBJECT-ORIENTED PROGRAMMING HAS ALSO BECOME MORE COMMON, FOR EXAMPLE IN C++ AND JAVA.

KEYWORDS-

- Identifiers Names in a programming language
- Keyword Has a special meaning in Java
- Java is a case-sensitive, free-format language
- Variable A named location in memory Must be declared before it can be used.

SOME TYPES OF KEYWORDS-

<code>int</code>	a positive or negative integer quantity
<code>double</code>	a positive or negative real number
<code>char</code>	a character (a single keyboard character, such as 'a')

Statements Types-

- Input/output statements
- Input statement collects a specific value from the user for a variable within the program
- Output statement writes a message or the value of a program variable to the user's screen or to a file
- Assignment statement assigns a value to a program variable

- Control statement directs the flow of control can cause it to deviate from the usual sequential flow

Managing Complexity

- Divide and conquer divide the problem into small pieces in a computer program
- Divide the code into modules or subprograms, each of which does some part of the overall task
- Empower these modules to work together to solve the original problem

Graphical Programming-

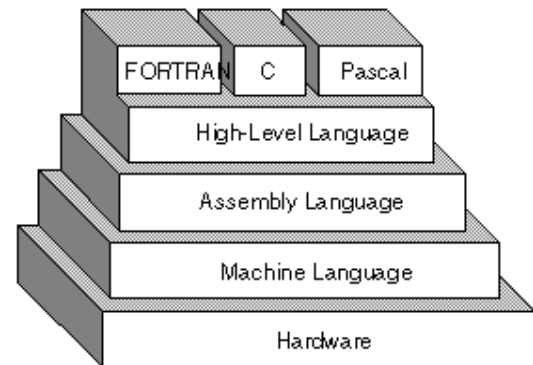
- Bitmapped display
 - The screen is made up of thousands of individual picture elements, or pixels, laid out in a two-dimensional grid
- Frame buffer
 - Memory that stores the actual screen image
- Terminal hardware displays the frame buffer value of every individual pixel on the screen

Computer Programming Languages-

- A programming language is an artificial language that can be used to control the behavior of a machine, particularly a computer
- Programming languages, like human languages, are defined through the use of syntactic and semantic rules, to determine structure and meaning respectively.
- Programming languages are used to facilitate communication about the task of organizing and manipulating information, and to express algorithms precisely.
- For 50 years, computer programmers have been writing code. New technologies continue to emerge, develop, and mature at a rapid pace. Now there are more than 2,500 documented programming languages!

Machine language-

- It is the lowest-level programming language



- Machine languages are the only languages understood by computers.
- While easily understood by computers, machine languages are almost impossible for humans to use because they consist entirely of numbers.

Assembly Level Language-

- An assembly language is a low-level language for programming computers.
- The word "low" does not imply that the language is inferior to high-level programming languages but rather refers to the small or nonexistent amount of abstraction between the language and machine language, because of this, low-level languages are sometimes described as being "close to the hardware."
- It implements a symbolic representation of the numeric machine codes and other constants needed to program a particular CPU architecture.
 - A utility program called an "assembler", is used to translate assembly language statements into the target computer's machine code.
- The assembler performs a more or less isomorphic translation (a one-to-one mapping) from mnemonic statements into machine instructions and data.

Example (C program to add 2 numbers)-

```
#include<stdio.h>           //header files
Void main()
{
    int a, b, c;           // declaration of 3 variables
    printf("Enter two numbers:\n");
    scanf("%d", &a);       // read 1st number
```

```
scanf("%d", &b);          // read 2nd number
c=a+b;                   // compute the sum
printf("Sum of 2 numbers is %d", c); //print sum
}
```

CONCLUSION-

- In a high-level language, the programmer
- Need not manage storage nor movement of data values in memory
- Can use more powerful program instructions that are more like natural language
- Can write a much more portable program

REFERENCES-

- www.google.com