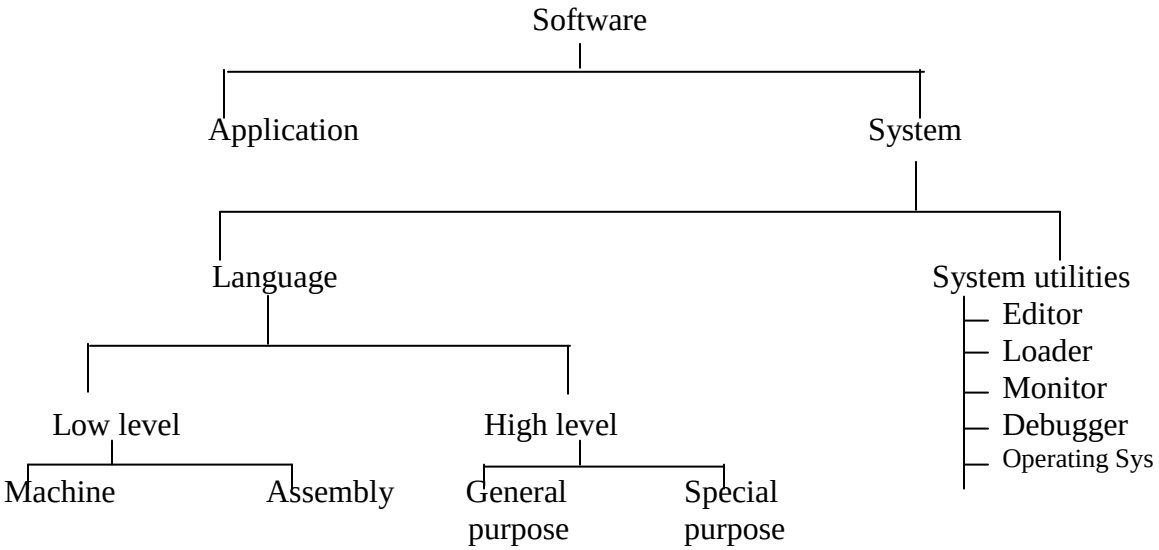
Software

A software or a program can be defined as a complete set of written instructions written by the programmers which enables the computer to obtain the solution of a problem.

Classifications:



Source code: The program written in high level/Assembly language.

Object code: The ,, ,, ,, Machine ,,

Translator: A program which translates a high level / Assembly language program into its equivalent machine code.

Three types:

1. Assembler: Assembly-machine
2. Compiler: High level-Machine (whole program at a time)
3. Interpreter: High level-Machine (one line at a time)

Application software: to perform source specific functions.

Ex: Word processing – MSWORD, WP, Spreadsheet- Excel etc.

System software: The set of programs, which provides the environment to write application program is known as system software.

Language: Low level language: Compatible with the hardware of the computer and consists of binary and machine codes.

Machine language: A language that is directly understood by the computer uses binary codes to represent operations and operand address.

EX-01111000 MOV A,B ; MOVE contact of rag B to Acc.

10000001 ADD C ; ADD contact of rag C to Acc.

01010111 MOV D,A ; MOVE contact of rag Acc. to D

Assembly language: A low level language in which symbolic codes (mnemonics) are used to code operations and alphanumeric symbols are used addresses.

High-level language: A machine independent programming language that uses English like syntax in which each statement corresponds to several lines of machine language instructions.

EX-General purpose-BASIC, PASCAL, C
Special purpose-LISP, COBOL, PROLOG

```
#include <stdio.h>
main()
{
printf("HELLO");
return 0;
}
```

Problem Solving Steps:

1. Problem Analysis
2. Algorithm Development
3. Program Coding
4. Program Compilation & Execution
5. Program Debugging & Testing
6. Program Documentation

Problem 1: Two numbers A and B are given. Find their sum.

1) Problem Analysis: Output –Sum
   Input- A, B

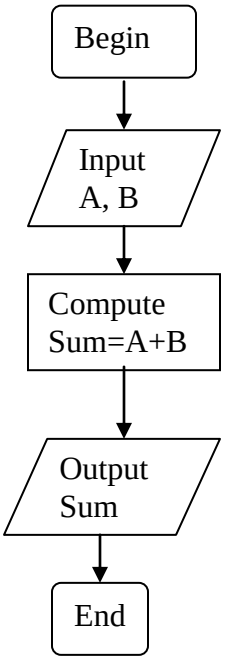
2) Algorithm Development: A finite sequence of instructions to solve a problem.

Algorithm:

1. Input A, B
2. Compute Sum = A+B
3. Output Sum
4. End.

Flowchart: Graphical representation of an algorithm.

Flowchart



3. Program Coding

```
#include<stdio.h>
#include<conio.h>
int main()
{
        int a,b,sum;
        printf("Enter two numbers");
        scanf("%d%d",&a,&b);
        sum=a+b;
        printf("The sum is %d",sum);
        getch();
        return 0;
}
```

5. Program compilation & execution:
    Compile- Alt+ F9 or Alt+C
    Execution – Ctrl+F9 or Ctrl+R
6.Debugging & testing:
    Debugging: a) Hand/ desk
    b) Syntax- compile time error
    c) Execution-run time error (divide by zero)
    d) Logical- A+B/2
    Testing: Sample data
7. Program Documentation
    Comments- /* …….*/ or //

Allocation of programmers time in developing a new program

1. Problem Analysis & Algorithm Development - 40%
2. Program coding - 20%
3. Debugging, testing & Documentation - 40%

100%

Problem 2: Given the temperature C in the centigrade scale. Write a program to find its value in the Fahrenheit scale.(Algorithm flowchart coding)

Problem 3: Write a program to find the largest number among the three numbers a, b and c. (Algorithm flowchart coding)