

UNIT-V

PROGRAM:- Is a set of or list of instructions that the computer must follow in order to process data into information. There are five steps in program execution.

1. problem Clarification
2. program design
3. program coding
4. Program testing
5. Program documentation and maintenance.

Problem Clarification: - In this step first user will understand the problem that means analysis of problem. This analysis is based on few considerations those are:

I. Clarify objectives & Users: - You need to write the steps for your objectives you are trying to accomplish. And you need to make sure the users of the program will be. Will they be people inside the company or outside, if you are preparing program for a company.

II. Clarify desired output: - Make sure you understand the output what the system designer want from the system. What information should be outputs include?

III. Clarify desired input: - once you know the type of output then you can think about input. What kind of input data is needed?

IV. Clarify desired Processing: - You should understand the procedure or process for executing any program or process of getting output from input.

V. Document the Analysis: - Throughout program clarification programs must document everything they do.

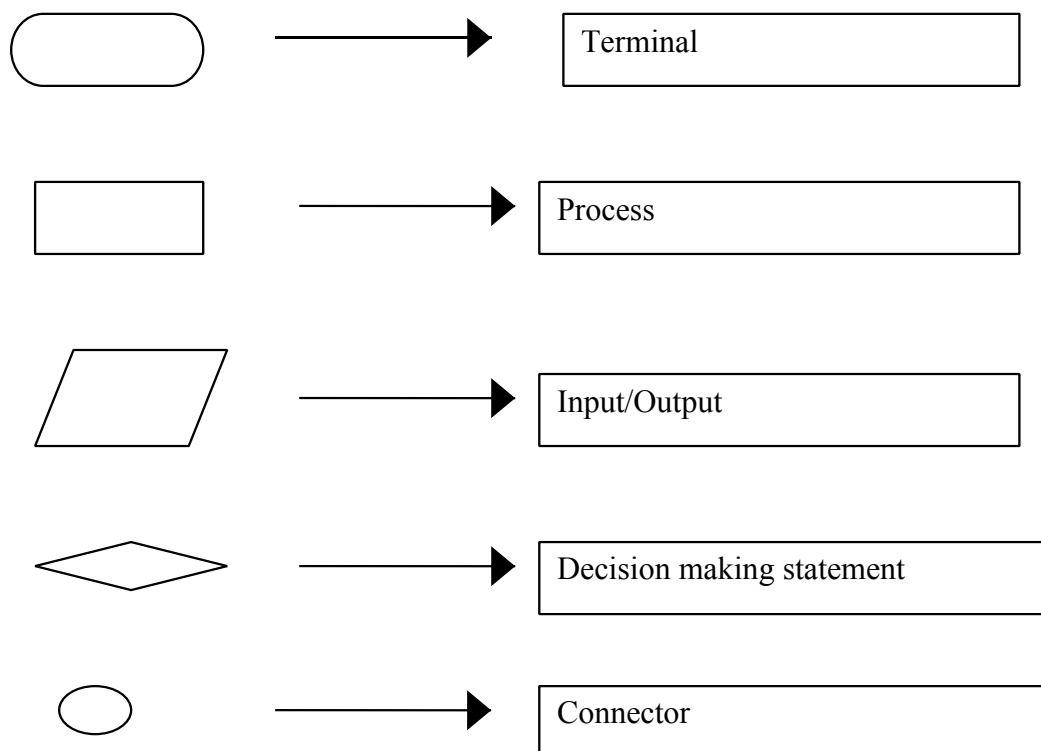
Program design: - In this program design step, the software is designed in 3 mini steps. First the program logic is determined through a top- down approach and modularization, using a hierarchy chart then designed in detail, either in pseudo code or graphical flow chart.

➤ **Determine the program logic using a Top- down Approach:** - A module is a processing step of a program. Such module is made up of logically related program elements. A hierarchy chart or structured chart illustrates the overall purposes of the program.

➤ **Design Details ,using Pseudo code or Flowchart :** -

Pseudo code: Algorithm, describing program by using some terms as If, then...else.

Flow Chart: Graphical representation of the program by using series of steps.



Terminal: Beginning & Ending of program.

Process: A calculation or assigning of work.

I/O: For giving input and taking output.

Connector: For connecting two symbols.

Control Structures:- when you are trying to determine the logic behind something you use words like “ if”, “ then” and “else”. There are three control structures.

1) Sequential Control

2) Selection Control

3) Iteration - loops

Program Coding: - Once the program has been developed, the actual writing the program is called coding.

1. **Select the appropriate Programming language:-** A programming language is a set of rules that tells the computer what operations to do.

Eg: COBOL, C,Java

Not all programming languages are suitable for all uses. Some are for mathematical use, some are for database. Thus choosing a language depends upon purpose of the program.

2. **Follow the syntax:-**

Syntax: The rules of the programming language.

Programming language have their own grammar just as human language.

Program Testing:- Running various tests and then running real world data to make sure the program works.

- **Perform Desk Checking:-** Desk checking is simply reading through, or checking , the program to make sure that it is free from errors.
- **Debug the Program:** - To detect, locate and remove all bugs/ errors in a program. Syntax errors are caused by typographical errors. Logical errors are caused by incorrect use of control structures.

Document & Maintain the Program: - Documentation consists of written descriptions of what a program is and how to use it.

- **Prepare user Documentation:** - When we buy a software package, you normally get manual with it. This is called user documentation.
- **Prepare operator Documentation:** - To help computer operator, programmers will provide operator documentation.
- **Write Programmer Documentation:** - one system is existing and few programmers are working, if these programmers are left then this programmer documentation helps the new comers to work with existing system.
- **Maintain the program:** - Maintain refers to any activity such as designed to keep programs in working condition, error free and up to date.

Five generations of programming languages.

A programming language is a set of rules that tell the computer what operations to do. The five generations of programming languages are as follows.

First generation: - Machine language (1947).

Machine language is the basic language of the computer, representing data as 1s and 0s. Machine language programs varied from computer to computer; that is they are machine-dependent.

The language is more technical, more flexible, less user friendly and faster.

Second generation: - Assembly language (1950)

Assembly language is a low-level programming language that allows a computer user to write a program using abbreviations or more easily remembered words instead of numbers. A language translator (assembler) is required to convert it to machine language.

A **compiler** is a computer program (or set of programs) that transforms source code written in a programming language (the *source language*) into another computer language (the *target language*, often having a binary form known as *object code*). The most common reason for wanting to transform source code is to create an executable program

In computer science, an **interpreter** normally means a computer program that executes, i.e. *performs*, instructions written in a programming language. An *interpreter* may be a program that either

1. executes the source code directly
2. translates source code into some efficient intermediate representation (code) and immediately executes this
3. explicitly executes stored precompiled code^[1] made by a compiler which is part of the interpreter system

Third generation: - High-level (early 1960).

A high-level or procedural language resembles to some human language such as English. Ex:- COBOL, C, C++, Fortran.

A language translator (compiler or interpreters) is needed to convert high level language into machine language.

Fourth generation:- Very high level or problem oriented language(1970).

A very high level or problem oriented language (4GL), are much more user oriented and allow users to develop programs with fewer commands. Ex:- RAD, DBMS, etc.

Fifth generation:- natural languages(1980).

It uses human language to give people a more natural connection with computers. Natural languages are part of the field of study known as artificial intelligence.

III Programming Languages Used Today:

1. FORTRAN: Stands for formula Translator, it is a language of mathematics and the first high level language. Actually designed for mathematical formulas, Scientific and Engineering Problem. Also useful for Complex Business Applications.

2. COBOL: Language of Business, adopted in 1960. Common business oriented language. Is most frequently used in business programming for language computers.

Adv's:

- It is machine independent
- Its English like statements are easy to understand even for non-programmer
- It can handle many files, records and fields
- Easy to handle input and output operations

Dis-Adv's:

- Because it is so readable it is wordy
- Thus simple programs are lengthy
- IT cannot handle mathematical operations, as well as FORTRAN

3. BASIC: Beginner's all purpose Symbolic Instruction Code

Used for creating and editing pages on the web. Can do anything from putting plain text on a webpage, to accessing and retrieving data from a database. Vary greatly in terms of power and complexity.

- **HTML**
Hyper Text Markup Language. The core language of the world wide web that is used to define the structure and layout of web pages by using various tags and attributes. Although a fundamental language of the web, HTML is static - content created with it does not change. HTML is used to specify the content a webpage will contain, not how the page functions. Learn HTML at our **HTML tutorials** section.
- **XML**
Extensible Markup Language. A language developed by the **W3C** which works like HTML, but unlike HTML, allows for custom tags that are defined by programmers. XML allows for the transmission of data between applications and organizations through the use of its custom tags.
- **Javascript**
A language developed by Netscape used to provide dynamic and interactive content on webpages. With Javascript it is possible to communicate with HTML, create animations, create calculators, validate forms, and more. Javascript is often confused with Java, but they are two different languages. Learn Javascript at our **Javascript tutorials** section.
- **VBScript**
Visual Basic Scripting Edition. A language developed by Microsoft that works only in Microsoft's Internet Explorer web browser and web browsers based on the Internet Explorer engine such as FlashPeak's **Slim Browser**. VBScript Can be used to print dates, make calculations, interact with the user, and more. VBScript is based on Visual Basic, but it is much simpler. Learn VBScript at our **VBScript tutorials** section.

- **PHP**
Hypertext Preprocessor (it's a recursive acronym). A powerful language used for many tasks such as data encryption, database access, and form validation. PHP was originally created in 1994 By Rasmus Lerdorf. Learn PHP at our **PHP tutorials** section.
- **Java**
A powerful and flexible language created by Sun Microsystems that can be used to create applets (a program that is executed from within another program) that run inside webpages as well as software applications. Things you can do with Java include interacting with the user, creating graphical programs, reading from files, and
- more. Java is often confused with Javascript, but they are two different languages. Learn Java at our **Java tutorials** section.

Software languages

Used for creating executable programs. Can create anything from simple console programs that print some text to the screen to entire operating systems. Vary greatly in terms of power and complexity.

- **C**
An advanced programming language used for software application development. Originally developed by Dennis Ritchie at Bell Labs in the 1970's and designed to be a systems programming language but since then has proven itself to be able to be used for various software applications such as business programs, engineering programs, and even games. The UNIX operating system is written in C.
- **C++**
Descendant of the C language. The difference between the two languages is that C++ is object-oriented. C++ was developed by Bjarne Stroustrup at Bell Labs and is a very popular language for graphical applications.
- **Visual Basic**
A language developed by Microsoft based on the BASIC language . Visual Basic is used for creating Windows applications. The VBScript language (also developed by Microsoft) is based on Visual Basic.

Java

A powerful and flexible language created by Sun Microsystems that can be used to create applets (a program that is executed from within another program) that run inside webpages as well as software applications. Things

Assembler: - It is program that translates the assembly language program into machine language.

Compiler:- A compiler is a language translator that converts the entire program of a high lever language into machine language before the computer executes the program.

Interpreter: - An interpreter is language translator that converts each procedural language statement into machine language and executes it immediately, statement by statement.

Source code: - the programming instructions of a procedural language are called the source code.

Object code:- The compiler translates source code into machine language, this machine language is called object code.

Pseudo code :-An outline of a program, written in a form that can easily be converted into real programming statements. Pseudo code cannot be compiled nor executed. The benefit of pseudo code is that it enables the programmer to concentrate on the algorithms without worrying about all the syntactic details of a particular programming language.