

Unit 6: Computer Software

Introduction

Collectively computer programs are known as computer software. This unit consisting of four lessons presents different aspects of computer software. Lesson 1 introduces software and its classification, system software which assists the users to develop programs for solving user problems is presented in Lesson 2. Many programs for widely used applications are available commercially. These programs are popularly known as application packages or package programs or simply packages. Advantages of package programs and brief outline of popular packages for word-processing, spreadsheet analysis, database management systems, desktop publication and graphic and applications are discussed in Lesson 3. Tasks for developing computer programs and brief introduction to some common programming languages are presented in Lesson 4.

Lesson 1: Introduction and Classification

1.1 Learning Objectives

On completion of this lesson you will be able to

- understand the concept of software
- distinguish between system software and application software
- know components of system software and types of application software.

1.2 Software

Software of a computer system is intangible rather than physical. It is the term used for any type of program. Software consists of statements, which instruct a computer to perform the required task. Without software a computer is simply a mass of electronic components. For a computer to input, store, make decisions, arithmetically manipulate and output data in the correct sequence it must have access to appropriate programs. Thus, the software includes all the activities associated with the successful development and operation of the computing system other than the hardware pieces. Hardware does the actual computing, and the software drives the computer. Fig. 6.1 shows the communication between users and hardware. Software plays the role to connect the users and the hardware. The application software convey the user intention to system

Software consists of statements, which instruct a computer to perform the required task.

software and the system software transfers and translates the intention for the hardware.

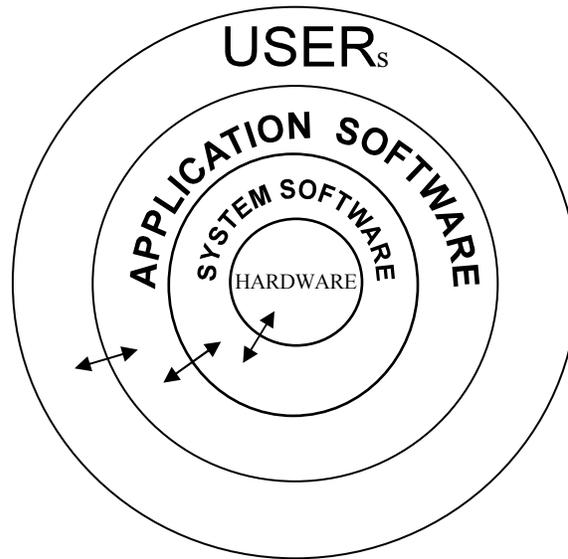


Figure 6.1 User and hardware communication.

Software activities cover three major areas : application programs, systems programs, and documentation.

Application programs perform a specific, well-defined task for a particular application.

Software activities

Systems programs consist of programs associated with a computing system other than application programs. Systems programs are usually available from the computer manufacturer as a part of the overall system. They aid in the creation of application programs.

Documentation includes everything put down on paper, such as a statement of the problems, flowcharting and coding, instruction sets, procedures, and the like. It plays an important role in software development.

1.3 Classification of Software

Software is generally classified as:

- (i) System software
- (ii) Application software

Computer Software

System software facilitates the use of the computer by the user.

Fig. 6.2 shows classification of computer software. System software consists of programs which facilitate the use of the computer by the users. These programs, perform such standard tasks as organizing and maintaining data files, translating programs written in various languages to a form acceptable to the hardware, scheduling jobs as well as aiding in other areas of computer operations. On the other hand, application software includes programs designed to perform specific user applications.

Application programs can be developed by the users themselves using suitable programming languages. There are many programming languages which allow us to write computer instructions in a text form. BASIC, FORTRAN, COBOL, PASCAL, C, C++, Matlab Java are some well known programming languages. However, writing application programs using such languages requires expertise and experience in programming and only experienced programs can write good quality application programs.

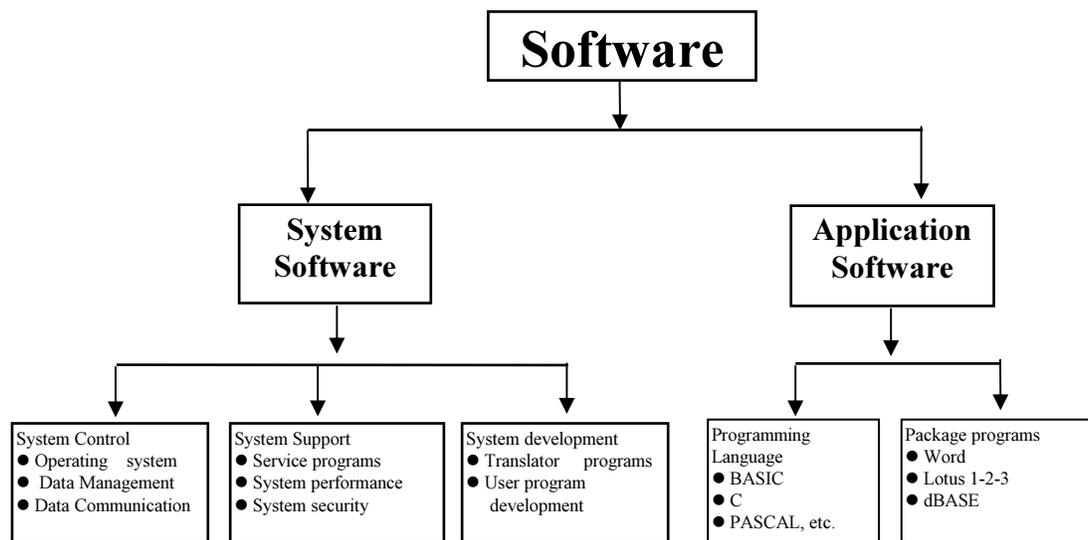


Fig. 6.2 Classification of software.

Application packages

Many application programs are now available commercially. Such programs, developed by experienced programmers can be readily employed for user applications. These programs are popularly known as application packages or simply packages. MS Word, WordPerfect, LA Tex and WordStar for word-processing, Lotus 1-2-3, and Excel for spreadsheet analysis and dBASE, FOXPRO, MS Access and ORACLE for database management are some popular application packages.

1.4 Exercise

1. Questions for short answers

- a. Define the terms: software, system software, application software and package program.
- b. Distinguish between hardware and software.
- c. List the names of at least five package programs.

2. Analytical questions

- a. What do you understand by the term software? Critically discuss classification of software.
- b. Write short note on application software.

Lesson 2: System Software

2.1 Learning Objectives

On completion of this lesson you will be able to :

- understand the concept of system software
- know the function of language translators
- understand the importance of testing, debugging and diagnostic programs.

2.2 Systems Software

Systems software assists the users to execute, generate, debug, test, modify and execute application programs.

Systems software refers to programs that assist the users to generate, debug, test, modify application programs, and then to execute them. These programs are generally written by computer manufacturers or software developers. The system programs can be used by different users and different application programs.

The most important component of system program is the operating system. It is a collection of program modules. Collectively they manage all the hardware, provides user-interface and some other user facilities. Other system programs include language translators (i.e. assembler, compiler, interpreter), text editors and some utility programs.

Machine Language

The machine language is written in words comprising of 1s and 0's.

The machine language is written in words comprising of 1s and 0s. Computers execute commands or instructions fed into the memories in machine language. The program is entered through the switches located on the front panel or the operator console. The switches are manually set to correspond to the binary code of each instruction; they are then stored in the program memory. After loading the program, the address of the first instruction is manually loaded into the program counter from the front panel. The program is then executed by pushing on start button. Programming in machine language has the following disadvantages:

Disadvantages of machine language.

- The program must be written entirely in machine language (i.e., with 1s and 0s).
- Manually setting the switches on the front panel, corresponding to each instruction in the program, is a tedious and cumbersome process and likely to introduce many errors.
- Errors detection and correction is tedious and consumes even more time.
- Programs written in machine language for a specific machine cannot be used for another type of machine. Rewriting the same

program for another machine is a laborious and time-consuming process.

Assembler

Assemblers are programs designed to translate assembly level codes into equivalent machine codes.

Many shortcomings of machine language programming are overcome by programming in assembly language. Assembly instructions of a computer have mnemonics associated with them. The assembly program is written using these mnemonics. All computers, have assemblers which are programs designed to translate assembly level codes into equivalent machine codes. For programming in assembly language, it is not necessary to write programs in absolute binary, octal or hexadecimal notations.

Assembly Language

Assembly Language

Programming in assembly language is easier than programming in machine language. The programmer is required to write a source code for each instruction in the program and the assembler translates it to machine language. The programmer maintains total control of the computer operation. Assembly language offers the greatest degree of efficiency in operation and execution of programs.

The Compiler

High level languages are widely used. They are procedure-oriented languages. They have been developed to enable the users to program the machines in a language similar to the language of the users. For example FORTRAN and BASIC, Pascal, C, C+ use scientific notations and subroutines in programming formats those are similar to mathematical formulas and equations. On the other hand, COBOL contains business jargon and emphasizes items such as interest rates and other terms widely used in business.

A Compiler translates programs written in high-level language into machine language.

A Compiler is a program that translates programs written in high-level language into machine language. Unlike the assembler, the compiler generates several machine language instructions for each source statement. A compiler can be readily modified and made to interface with several different computers. It is not written for any specific computer. A user can write programs that can be adapted to several computing systems. Thus, the user can formulate problems efficiently without precise knowledge of the computer architecture. Compilers are complex, long programs compared to assemblers and they require more storage space in the memory than assemblers.

The Interpreter

An assembler and a compiler perform an intermediate function in the translation from the source program to the object program. An assembler or a compiler produces the object program, which is loaded into the computer memory and executed by the computer.

The interpreter translates and immediately executes each instruction, of the source program.

The interpreter is a translator program that differs from the assembler or the compiler on several significant points. The interpreter does not prepare an object program. It translates and immediately executes each instruction, of the source program. Thus the interpretive language is an interactive language; it enables us to load one instruction into the computer at a time and have it translated and executed. This process allows the programmer to check the results immediately. The interpreter is a debugging tool and is useful during the program development stage. In the interpretive mode the execution time of the program is extended, but the program developmental effort and the program development time are decreased.

The Text Editor

While developing a program, it is often necessary to edit it. Editing involves making corrections or modifications to instructions, deletions, and additions of instructions, sequencing instructions etc. A special system program called the editor greatly simplifies the editing tasks. The editor enables the user to retain the program in the memory and modify them as required by specifying the changes to be made. A portion of the memory, called a buffer, is reserved for editing. Lines of instruction are entered from an external I/O device and stored in the buffer by the editor. The editor generally assigns a number to each such line and to each correction or modification. The editor automatically corrects line numbers of instructions or data lines. The edited text can be printed by a printer or recorded on suitable I/O devices.

The editor makes it possible to retain the program in the memory for modification by specifying the changes to be made.

Testing, Debugging and Diagnostic Programs

Programs created by us are prone to error. It is not realistic to expect a newly created program to run correctly the first time. Before a program can be put into productive use it must be debugged. A bug is a defect or imperfection in a program. In the context of programming it is an error in a program that is responsible for undesirable results. Testing of a program involves validation of program design. Thus testing involves data that are intentionally used to reflect the worst-case situations.

Testing, Debugging and Diagnostic Programs.

Computer Basics

In order to assist the programmer in locating and rectifying programming errors or bugs, several tools are available. These tools could be hardware aids such as logic analyzers or software aids such as debug programs. A debug program is loaded into the memory with the object program that is to be debugged. If errors occur during run, the debug program is activated to identify, locate, and correct the error. A debug program generally includes several features that aid the programmer in debugging tasks. Some of the common features are described below.

Some common feature of debuggers.

Display Register Contents: This feature allows the programmer to check contents of certain registers of the CPU during the execution of the program. The contents of the registers can be printed or displayed on appropriate devices. It is also possible to modify the contents of certain registers if necessary.

Display/Replace memory Contents: The programmer can access the contents of a memory location and display it. The contents can then be modified if necessary.

Memory Dump: This feature makes it possible to print the contents of a specified group of memory locations. Thus a programmer can view the contents of desired memory locations and determine accuracy of the data. A line printer or CRT display is required for this.

Output Memory Content: The contents of a memory are transferred and stored in some other mass storage devices such as disks or magnetic tapes.

Breakpoints: Breakpoints are interrupts in the object program. They are used to examine the current status of certain critical CPU registers, memory locations, or I/O ports. When the execution of the object program reaches a breakpoint, its execution is temporarily halted. The debug program then allows the programmer to examine and alter the contents of the selected register, memory locations, or I/O ports to correct errors.

Disassembly: This is the reverse of the assembly process. The operation consists of reading out the contents of the program memory into assembly language.

2.3 Exercise

1. Questions for short answers

- a. Distinguish between machine language, assembly language and high level language.
- b. What is the function of a language translator?
- c. Distinguish between assembler and compiler.
- d. Give the main characteristic of an interpretive language.
- e. List the main features of a debug program.
- f. List the main functions of a text editor.

2. Analytical questions

- a. What is a language translator? Discuss the characteristics of different types of language translators.
- b. Discuss the disadvantages of machine language programming.
- c. Distinguish between compiler based languages and interpreter based languages. Give one example of each type.
- d. What is a debug program? Discuss the major features of a debug program.

Lesson 3: Application Package Programs

3.1 Learning Objectives

On completion of this lesson you will be able to :

- know the types and characteristics of application packages
- know the names of common application packages
- understand the features of some popular packages.

3.2 Introduction

Some advantages of packages.

Package programs are developed by experienced programmers and marketed by software vendors for widely used applications. The package generally achieves the goals of lower cost and reduced risk of errors. Whenever feasible, the use of an appropriate application package should be considered as an alternative to program development.

Integrated package

Integrated packages combine several applications into a single program packages. Popular examples of integrated packages are MS Office, Microsoft Word, MS Excel, MS Power Point, MS Access, WordPerfect, Lotus 1-2-3 and Symphony. The packages provide several standard applications such as word processing, spreadsheet analysis and databases. Data can be easily exchanged between the programs of the integrated package. As an example it is easy to write a form letter using word processor and the address it to a group of names prepared by a word processor package.

3.3 Examples of Package Program

Popular packages.

There are many different kinds of programs available in the market to solve user problems. These are known as package programs. Package programs are very popular and widely used in different disciplines. The main categories and their examples are given below:

Categories	Popular Package Programs
Word Processors	Word, WordStar, WordPerfect, Latex etc.
Spreadsheets	Lotus 1-2-3, Excel, Quicken, QuattroPro, SuperCalc.
Data Management	dBASE, ORACLE, FoxBASE+, Idealistic, SuperBase Professional, MS Foxpro, MS Access etc.
Integrated Software	PFS: First Choice, Microsoft Works, Enable/OA, WordPerfect, MS Office etc.
Desktop Publishing	PageMaker, Venture Publisher, Professional page, PFS: First Publisher, Express Publisher, Publish it! etc.

Computer Software

Graphics	PC Paintbrush, CorelDRAW, Harvard Graphics, Freelance Graphics etc.
CAD	AutoCAD, Generic CADD, ORCAD, MiniCAD plus.
Communications	PROCOMM PLUS, CrossTalk, PFS: First Choice (Communications), Apple Works GS, MicroPhone 11, White Knight, SmartCom 11 etc.
Educational	World Atlas, The Human Brain, Spelling Bee, Fantavision, HyperStudio etc.
Recreational	Indy 500, Chuck Yeager's Advanced Flight Trainer, Monopoly, Hard Ball, Omar-sharif etc.

3.4 Advantages of Package Programs

Advantages and benefits of using some packages for popular applications are briefly described here.

Word Processing

Features of Word Processing packages.

- Document can be typed and then stored on a disk for later use.
- Easy text entry and editing.
- On-screen formatting: such as bold italic, underlining, and highlighting etc.
- Most word processors have spelling checker, thesaurus, and grammar checker to help to improve the accuracy of writing.
- A variety of print options to customize the program for printer.
- The capability to change the font and style of texts.
- The capability to interface with graphics and other software packages.

Spreadsheet Analysis

Features of spreadsheet packages.

- Data entry can be automated and checked for errors.
- Easy editing
- The column-and-row formats are easy to understand and use.
- Formulas perform calculations and are saved with spreadsheet.
- One can sort, copy, and move columns and rows with formulas intact.
- One can print a spreadsheet in a variety of formats.
- Built-in graph generator to create graphs from the data in a spreadsheet.

Data Management

Features of Data Management packages.

- One can reuse the data, access data easily, edit, sort, search, and arrange the data, create report, print labels etc.
- One can arrange data easily by using a database program's search capabilities.
- Printing features enable to generate mailing labels and reports.

Integrated Software

Features of Integrated Software.

- Data are compatible among applications. As an example, one can transfer data from database or spreadsheet to word processor or vice versa.
- The menu system is generally the same, lessening the number of menus and options.
- Having an all-in-one package means that less time is spent for loading, opening and closing files.
- Integrated software costs less than the same number of applications purchased individually.

Desktop Publishing

Features of Desktop Publishing

- On-screen formatting and layout facility of text and graphics.
- The capability of assembling documents quickly and making changes easily.
- Import text from any word processor or type text directly in the program.
- Print on a variety of printers, from dot-matrix printers to postscript laser printers.

Graphics and CAD

Features of Graphics and CAD

- One can use the art to create over and over again without losing the original quality.
- With on-screen graphics, it is not required to go back to the drawing board each time.
- One can make small modifications to the original art until achieving the desired design.
- One can create a variety of charts easily with presentation graphic software.
- Most paint programs provide a variety of shading and filing techniques.
- The art tools in a graphics program give control over the precision of creations.
- Draw programs produce graphics in postscript printer format.
- With many graphics programs one can import art directly into popular desktop publishing programs.

3.5 Popular Package Program

Microsoft Word

Microsoft Word- a powerful word processor.

Microsoft Word (MS Word) possesses a wide range of powerful and easy-to-use features. Word is popular among PC and Mac users. With Microsoft Word, one can integrate text with graphics and revise information within a document. Microsoft Word has special features such as print preview, side-by-side columns, built-in outlining, and automatic pagination. Word for Windows allows to open multiple windows on the currently open document or more than one document at one time. Word is designed to help print reports, memos, and documents and performs all the basic word processing techniques, such as deleting, italicizing and highlighting. Word also has flexible tabs to produce great-looking tables and can mix fonts and font sizes easily. Microsoft Word has spelling checker, thesaurus, equation editor, sorting facility, drawing facility, spreadsheet facility to generate graphs and an undo command. Many users prefer Word because of its compatibility to other systems. Using word it is possible to share data between PC and Mac systems. It is also compatible with WordPerfect for Windows, PageMaker and many other windows based software.

WordPerfect

Features of WordPerfect

WordPerfect (WP) is another most popular word processing package. A full-featured word processor with an extensive range of features, WordPerfect appeals to users who demand many sophisticated tasks from their word processor. Easy text entry, powerful thesaurus and spell-checking routines, support for a wide range of fonts and printers, capability to create customs dictionaries, on-screen formatting, print preview, graphics interfacing and the column creating facilities are the main features of this program. WordPerfect also has a master document features for assembling one large document from a series of subdocuments, and a tutorial to help the inexperienced users. One also can use WordPerfect to program, to plan, and to draft writing projects. WordPerfect for windows is available for microcomputer users.

Microsoft Excel

Microsoft Excel - a popular spreadsheet package.

Microsoft Excel is one of the most popular spreadsheet packages originally available only for the Macintosh. Microsoft - Excel for other microcomputers works with Microsoft Windows. Many operations that require typing in other applications can be handled with the mouse in the Excel. It also has options for producing a variety of reports in different fonts, a feature usually limited-or unavailable in many other spreadsheets packages.

Lotus 1-2-3

Lotus 1-2-3- a popular spreadsheet package.

Another most popular spreadsheet package is Lotus 1-2-3. It combines three main features: spreadsheet, database and graphics. With lotus 1-2-3, users can create simple or complex spreadsheets, use the spreadsheet as a database, or graph spreadsheet information. A wide range of people use Lotus 1-2-3. From accounts to financial planners, business owners, and managers, Lotus 1-2-3 is the spreadsheet package of choice. 1-2-3's many features include an easy-to-use spreadsheet layout, an understandable control panel, a wide range of functions, easy printing procedures, graph creation and printing and macro capability. Lotus 1-2-3 for windows is available.

dBASE

DBASE - a popular database management system.

The dBASE is an integrated software package; which has menu oriented database processing and programming facilities. It is a data manager; a piece of software that lets the user has freedom in the conceptualization and creation of database for all types of business applications. Since business depends on timely information dissemination, the value of a powerful programmable utility for database generation, maintenance, and query cannot be overstated. The dBASE can be executed on a variety of microcomputers, under any one of the popular operating systems. Different version of dBASE is known as dBASE III and dBASE III +, dBASE IV etc.

FoxBASE+

FoxBASE+

FoxBASE+ is a popular relational database program available for microcomputers including the Macintosh. Similar to the popular dBASE, FoxBASE+ is a user-friendly, intuitive program that operates faster than many other database programs. This program enables us to create forms, link databases, enter data, and design custom reports easily. FoxBASE+ can be used to add special touches to databases by changing colour, font, size and style settings.

Microsoft Works

Microsoft Works combines word processing, spreadsheet and database.

Microsoft Works is an integrated program that combines: word processing, spreadsheet, database and communications. Microsoft Works has enthusiastic following in both the PC and the Mac worlds. Because of easy features and friendly user interface, the applications in Microsoft Works are easy to understand and use. Its word processor enables one to enter, edit, save, and print texts and reports. Specialized search-and-replace features, thesaurus and spelling checker also help to ensure the accuracy of work. Using Microsoft Works spreadsheet, one can create

Computer Software

simple or complex works, perform what-if calculations, and use formulas to take the drudgery out of bookkeeping chores. The data manager gives a great deal of flexibility in generating data-entry forms, working with fields, displaying data, and generating reports. The communications module accesses the world outside computer by transferring files to and from remote computers. Finally, the features in Microsoft Works can be used to format, merge data between applications, and exchange information with other programs and modules.

PFS: First Choice

PFS: First Choice, one of the most popular integrated packages for microcomputers combines the following applications:

- Word processing
- Database management
- Communications
- Spreadsheet
- Graphics.

It has a multitude of users from sophisticated business managers to the hobbyists. Some users include business people who can type correspondence, balance books, and manage files, and business owners and managers who can analyze and maintain information. Other users include support staff working in data entry, creating visual presentations, or producing reports. PFS: First Choice automates many business tasks traditionally done by hand. With this program, one can perform accounting tasks, such as analyzing data and creating and printing spreadsheets. The word processor can be used to type correspondence, produce manuals, print documents, create presentations, and so on. PFS: first Choice also enables one to graph data, print graphs, plot graphs, create slide shows, and perform basic data-entry tasks. With the database management part of PFS: First Choice, one can design database forms, create mailing labels, edit and analyze data, update files, retrieve files, and create and print database reports.

PFS: First Choice combines several applications.

PageMaker

PageMaker is a most popular desktop publishing software package for microcomputers. Also called a page-composition program, PageMaker enables one to combine text and graphics on a page to compose documents. PageMaker accept text from a variety of popular word-processing programs and graphics in popular file formats. PageMaker can be used to format a page in a number of columns and formats, use different fonts, import sophisticated graphics, create on-screen graphics,

PageMaker is a high end desktop publishing program.

Computer Basics

create master pages, and perform many high-end functions. Although it is a high-end desktop publishing program, many people find the program easy to learn and use. Used with Microsoft Windows, PageMaker has familiar menus, pop-up dialog boxes, and a Mac-like display that most find easy to understand and use.

Harvard Graphics

Harvard Graphics is a popular presentation graphics program for PC users. Used primarily in business, Harvard Graphics is a feature-laden program with many enhancement options, but most users can find their way around easily. A comprehensive help system assists in learning the use of the software. Few features of the program are:

Features of Harvard Graphics.

- The Chart Gallery, offering several chart types in a variety of formats.
- The Draw utility, which can add customize art to charts.
- The ability to put charts in an automated sideshow.
- The choice of animating arts giving the illusion of real animation in presentations.
- Supports for data imported from a variety of spreadsheet programs, such as Lotus 1-2-3 and Microsoft Excel.
- Output on display, printer, plotter, or film recorder devices.
- Templates and macro creating facility.

AutoCAD

AutoCAD is one of the best CAD programs in the market. It offers the following features:

Features of Auto CAD

- An easy-to-use interface with pull-down menus and dialog boxes.
- Support for a mouse or digitizer.
- Capability to layer, rotate, copy, mirror, move, stretch, and scale.
- 3-D wire frame modeling.
- Up to 16 different views.

The version of AutoCAD which can be used with Microsoft Windows is easy-to-use.

3.6 Exercise

1. Questions for short answers

- a. Distinguish between single application and integrated packages.
- b. List popular desktop publishing packages and database management packages.
- c. List the features of Microsoft word.

2. Analytical questions

- a. What is the integrated package? Discuss its features and advantages. Give one example with its capability.
- b. What is a graphic package? Give at least two examples of graphic package and discuss the characteristics of any one of them.

Lesson 4: High Level Languages and Software Development Cycle

4.1 Learning Objectives

On completion of this lesson you will be able to :

- understand high level languages
- software development cycle.

4.2 Programming and Languages

A Computer does not have a brain like ours and it cannot do anything by itself. A computer cannot understand human language, so an instruction for a task has to be told in a special computer language. One can give an instruction to a computer which is to be carried out straight away. It is also need to store a set of instructions which are to be carried out over and over again or at a later time.

A Program is a set of instructions

A set of instructions written in a computer language is called a program. Writing a program is called programming and the languages used are called programming languages. A program can be stored on a disk, so that it is available to be run when required.

Inside the computer, operations are done using electronic pulses. A special program translates instructions into electronic pulses that the computer can understand. In a program the instructions contain special words and symbols and the order of the words is very important. The pattern of electronic pulses for an instruction is called machine code. One could write instructions directly in machine code but it is very difficult. In assembly language instructions are given by short mnemonic forms, called assemble mnemonics. High level languages are much easier to understand and quicker to use than assembler mnemonic or machine code. Instructions of a program written in high level languages are translated by computers, in machine language with the help of the compilers before execution.

High Level Languages

There are several high level programming languages in use. Following computer languages are popular and have widespread applications :

- BASIC
- C
- FORTRAN

Computer Software

- PASCAL
- COBOL
- C++

BASIC

BASIC : It is the short name for Beginner's All purpose Symbolic Instruction Code. It is a comparatively easy language for beginners. Initially BASIC is the commonly used language for microcomputers. It was developed by J.G. Kemeny and Thomas Curtz of Dartmouth College of USA in 1964. Many different versions of BASIC are in use. One is called standard BASIC, but this is so elementary that it is not used now. Most microcomputers use a BASIC devised by a company called Microsoft. QBASIC supplied with the microcomputers is popular now. BASIC has several advantages over the other computer languages they are:

- It is easy to learn and use.
- It presents several powerful additional facilities not available in many other languages.
- It is designed for interactive use rather than batch processing mode. The user can correct errors as and when they arise during program execution.

The language is easily adaptable on all computers. BASIC is easy and suitable for non-professionals like students.

C

C : Dennis Ritchie, a programmer at AT &T Bell Laboratories in USA designed the C language in the mid 1970s. C allows the programmer to write code with low-level access to the hardware but with high-level structured programming concept. C is a popular language. C proved suitable for writing programs that had previously been written exclusively in assembly languages. AT & T UNIX operating system was written in C. The language is portable to microcomputers. Most PC applications and system programs are written in C.

C++

C++ : C++ is an object oriented programming (OOP) language developed by Bjarne Stroustrup and is an extension of C language. C++ considered to be an intermediate level language, as it encapsulates both high and low level language features. Initially it was called "C" with classes, as it had all properties of C language with concept of classes. Then it was renamed to C++ in 1983. The main highlight of C++ is a collection of predefined classes. The main features of C++ are: polymorphism, inheritance, virtual and friend functions, namespace and pointers. C++ is one of most popular programming language utilized with system/application software development, client server application and to make CGI scripts.

FORTRAN

FORTRAN : Fortran stands for Formula Translation. It was the first programming language developed by IBM in 1957. It has been modified

many times and the later versions were given the name like FORTRAN II and FORTRAN-IV. The most popular version which was developed in 1966. Latest additions are FORTRAN 77 and FORTRAN 90. FORTRAN syntax is very rigid. It is popular in engineering and scientific calculations.

Pascal

Pascal: Pascal is popular on small computers. Pascal was designed and developed by Nicholas Writhe to teach computer programming to students. It belongs to ALGOL family. A Pascal program consists of nested blocks for structured programming.

Cobol

COBOL: It stands for Common Business Oriented Language. It is a popular business language for data processing. It was designed in late 1960s with a view to provide a common language for business applications. COBOL is independent of any specific computer system, and thus a program written in the language can be transferred to another computer with marginal changes only.

Some Other High Level Languages

Some other High level languages

ALGOL: is a short name for Algorithmic Language. Originally developed in 1958, it is more precision language than FORTRAN for scientific and numeric calculations. **Forth:** is a powerful language designed only for fourth generation computers. Useful in home computing. **RPG:** Stands for Report Program Generator. It is a commercial language specially developed for reporting.

ADA: Designed by U.S. Defense Department for its real-time applications and is also suitable for parallel processing. **PL/I:** (Programming Languages/I), which combines the features of both COBOL and FORTRAN and can be used for scientific as well as business applications. It was introduced in the late 1960s by IBM. Its application is limited to only IBM computers.

Snobol: was developed in 1962. It is quite different from other languages in the sense that it can be used for processing messages, patterns and strings. **LISP:** stands for List Processor. It is an artificial intelligence programming language.

4.3 Software Development Cycle

Software development for a computer system involves a number of specific tasks. They are shown as a flowchart in Figure 6.3. The tasks are explained below.

Problem Statement and Specification: This is the starting point of all succeeding activities. In this step the problem is defined and the tasks to be performed are outlined and specifications are written. The specifications should be clear, concise, and well written so that the users of the software readily understand them. The inputs and expected outputs are specified along with other constraints of the program. Procedures for handling errors should also be included in this step.

Tasks for software development

Program Design: This phase presents a plan of attack for solving the problem. Several techniques (e.g., modular programming, top down design, structured programming) are available for this. Flowcharting is a convenient starting point. After the overall plan is established, the designer selects suitable algorithm or a set of procedures for the job.

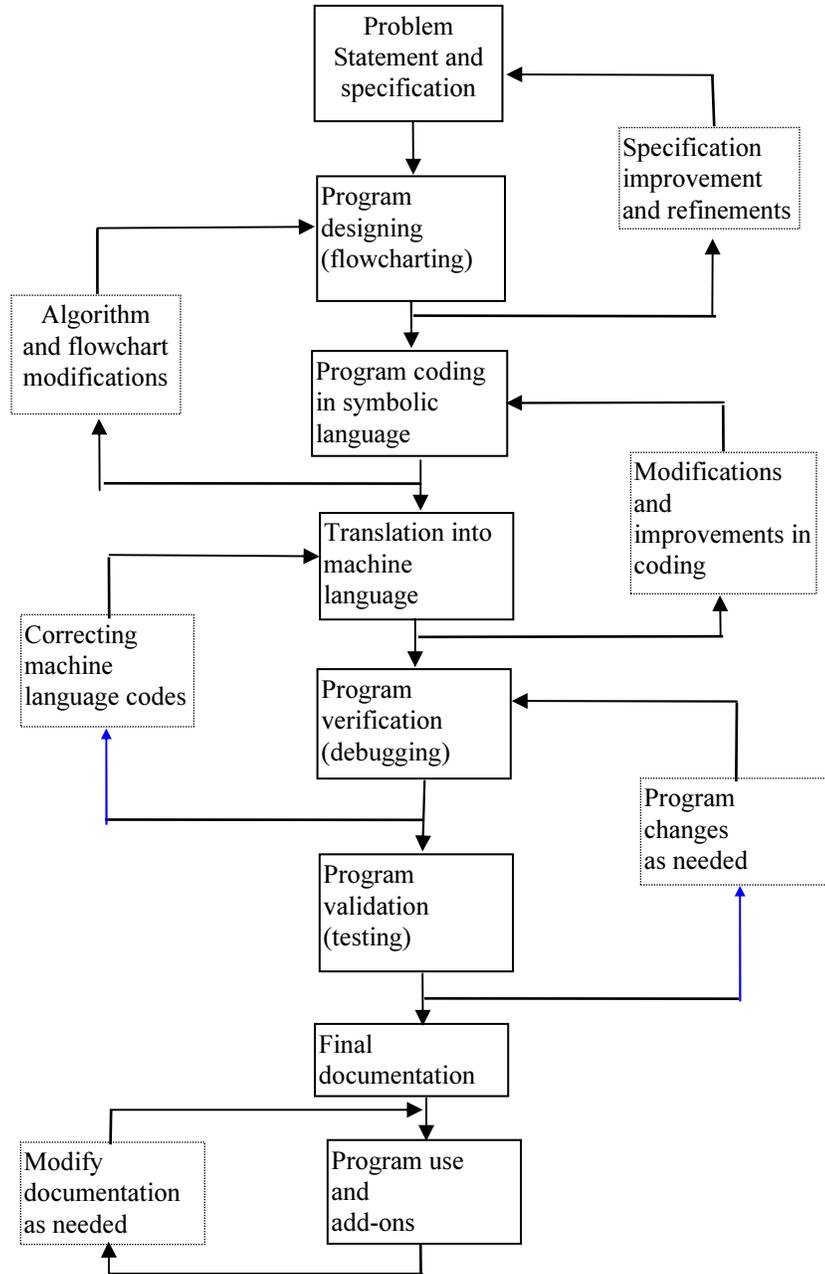


Figure 6.3 Computer software development cycle.

Computer Basics

Program Coding: This step consists of describing the program into a symbolic language (BASIC, C, Pascal etc.). The program produced in a symbolic language is called the source program. The source program is translated to machine language before execution by a computer. The program produced in machine language is called the object program. A source program may be in either assembly language or in a high level language.

Translation into Machine Language: The CPU understands the machine language. Therefore, the source program must be translated into an object program in machine language and stored in the program memory before the computer executes it. The translation of the source program in assembly language is performed by a special program called the assembler, which effectively stores the binary equivalent of the assembly instructions in the computer memory. Thus, assembler produces the object program. The source program in a high level language is translated by another translator program, called a compiler.

Program Verification: In the initial stage, every program generally contains at least a few errors. In this step such errors are found and corrected. This phase of software development is time consuming and could be frustrating. Isolating the particular fault is called debugging. Fortunately, several tools are available to simplify the debugging process. There are software tools such as debugging programs and editors and hardware debugging tools such as logic analyzers.

Program Validation: The debugging state involves finding errors introduced during coding and or translation process. After the program is debugged, the next step is program validation. Validation involves testing the program to see if it satisfies the requirements and specifications. This process involves running the program with various inputs and checking the outputs.

Final Documentation: After the program is debugged and validated, documentation is prepared for end users and the maintenance people. Documentation includes specifications, requirements, flowcharts, memory maps and program listings. The documentation is also used by designers if the programs are to be modified or extended in the future.

Program Usage and add-ons: With time the user tends to change the ways in a particular application and this requires changes in the programs. Such changes usually improve the programs and tend to make them more sophisticated. Also, the user may require extending the program - beyond the tasks for which the program was initially designed. The documentation is required for such extension of the program.

4.4 Exercise

1. Questions for short answers

- a. Define the terms: high level language, source code and object code.
- b. Name the programs used for converting source code to object code.
- c. List the names of six high level languages.
- d. Write a note on programming language C.
- e. List the steps for program development cycles.

2. Analytical questions

- a. Name the three levels of programming languages and discuss the difference between them.
- b. Write a short essay on high level programming languages.
- c. Discuss the importance of program development cycle. List the steps of this cycle and briefly explain each step.

