Basic Computer Science

Version – 3.0

05th August 2003

Table of Contents

Day-1 Schedule

Session 1
Introduction
Number Systems

Session 2
Principles of Programming
Flow chart problem solving

Day-2 Schedule

Session 1
Computer Software
Files and Database Management Systems

Session 2
Flow chart problem solving

1 Introduction

The instrument of the present technological revolution is the computer and the purpose of this chapter is to raise your awareness of computers to a higher level and form the foundation for the technical chapters that follow.

There are three key characteristics that make computers an indispensable part of our lives: speed, reliability and storage capacity. By products of these characteristics include productivity, decision-making and storage capacity.

1.1 What is a Computer

A computer is an electronic device, which converts *data* into *information***.** It is a high-speed data processing machine, which accepts data, processes data and outputs information.

1.2 Computer Generations

"Generation" in a computer talk, is a step technology. The custom of referring to the computer era in terms of generations came into wide use only after 1964. There are totally 5 computer generations known till today. Each generation has been discussed below in detail:

1.2.1First Generation (1942 - 55)

ENIAC, EDVAC etc. machines and others of their time were made possible by the invention of the "vacuum tube", which was a fragile glass device that could control and amplify electronic signals. These vacuum tube computers are referred to as first generation computers.

1.2.2Second Generation (1955 - 64)

The transistor, a smaller and more reliable successor to the vacuum tube, was invented in 1947. However, they were used in computers only after a decade. The second generation emerged with transistors being the brain of the computer.

1.2.3 Third Generation (1964 - 75)

Advances in electronic technology continued and the advent of "microelectronics" technology made it possible to integrate large number of circuit elements into very small (less than 5 mm square) surface of silicon known as "chips". This new technology was called Integrated Circuits (ICs). The third generation was based on IC technology, and the computers that were designed with the use of integrated circuits were called third generation computers.

1.2.4 Fourth Generation (1975 Onwards)

Initially, the ICs contained only about 10 to 20 components. This technology was named SSI (Small Scale Integration). Later, with the advancements in technology for manufacturing ICs, it became possible to integrate up to 100 components on a single chip. This technology came to known as MSI (Medium Scale Integration). Then came the era of LSI (Large Scale Integration) when it was possible to integrate over 30,000 components onto a single chip. Effort is still on for further miniaturization, and it is expected that more than one million components will be integrated on a single chip, known as VLSI (Very Large Scale Integration.

1.2.5 Fifth Generation

Scientists are now at work on the fifth generation computers – a promise, but not a reality. They aim to bring us machine with genuine I.Q., the ability to reason logically, and with real knowledge of the world. Thus, unlike the last four generations, which naturally followed its predecessors, the fifth generation will be totally different and totally new.

1.3 Functional Components of a Computer

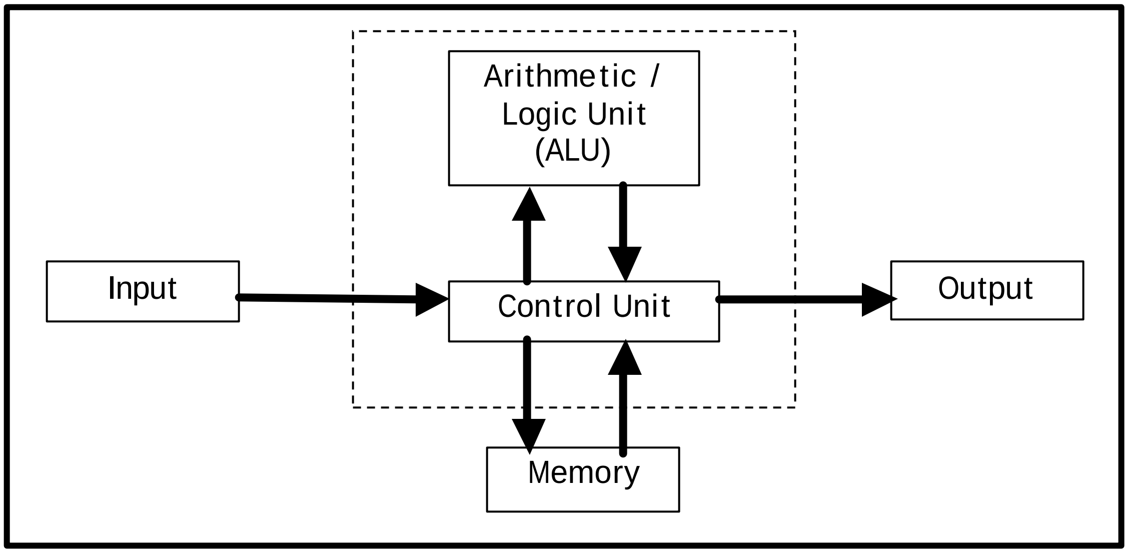


Figure 1-1 Functional Components of a Computer

1.3.1 Central Processing Unit

It is the part of the computer whose operations we cannot see. It is a highly complex and extensive set of electrical circuitry that executes stored program instructions. Consists of:

Arithmetic / Logic Unit (ALU)

Performs arithmetic and logical operations like addition, subtraction, comparison etc.

Control Unit (CU)

- Like an orchestra leader the control unit does not execute the instructions itself but directs other parts of the system like the ALU and memory.
- Co-ordinates the Operation of the Hardware.
- Co-ordinates the flow and execution of instructions and frequency with which instructions are executed

Memory Unit

It is that part of the computer that holds data and instructions for processing. Also known as primary storage, main storage or internal storage. Although closely associated with the CPU, technically memory is separate from it. The two basic types of memory chips in every computer are known as *random-access memory RAM* and *read-only memory ROM.*

ROM

Contains programs and data sometimes-called *firmware* that cannot be altered. It is *nonvolatile*-its contents do not disappear when the power is turned off. Specialized tools like ROM burners, called PROM chips or programmable read-only memory chips can be used to change instructions in ROM chips.

RAM

Has been described as the *computer's scratch pad*. Can be called *read-write chips*. User can usually read data stored as well as write data. RAM is volatile-that is, data is lost once power is turned off. The more RAM, the more powerful the programs you can run.

Both CPU and memory are on silicon chips, which are smaller than thumbtacks. CPU is on a chip called *microprocessor*. Microchips form the brain of the computer and they process information as a series of "*on-off*" electrical signals.

Types of printers:

- Character printers i.e. daisy wheel printer
- Line printers i.e. band printer, chain printer, drum printer
- Ink-Jet printers
- Laser

Secondary Storage Devices

- Magnetic Tape

Used for sequential storage. The tape has an iron-oxide coating that can be magnetized. Data is stored as extremely small, magnetized spots, which can be then read by a tape unit into the computer's main storage. The magnetic tape unit is about the size and shape of a refrigerator. It contains a read/write head and an erase head. Two reels are used, the supply reel and take-up reel. The tape is allowed to drop down into vacuum chambers that prevent the tape from breaking. Uses the concept of blocking to store records. *Blocking* consists of putting together logical records into one physical record. The *blocking factor* refers to the number of logical records in one physical record.

Magnetic tape memories are similar to the commonly used audio tape recorders. A magnetic tape drive consists of a spool on which a magnetic tape is wound. The tape is transported across a set of magnetic heads and is taken up on another spool. Between the two spools, a set of nine heads is mounted to write and read information on the tape. Each head operates independently and stores information along one track on the tape. The nine heads record information on nine parallel tracks, parallel to the edge of the tape. Out of nine tracks eight tracks are used to record one byte of data and ninth track is used to record the parity bit. The width of the tape is half inch. The recording density is defined as number of bits per inch length of the tape. This is typically 1600 bpi (bits per inch) X 9 = 14400 bits.

- Magnetic Disk Storage

It is a direct access storage device. Various types are hard disks, diskettes etc. A disk drive allows data to be read or written from and to a disk. Several disks are assembled together in a disk pack. Each disk has tracks and data is recorded as magnetic spots on the tracks. Tracks are concentric and each track contains the same amount of data. In the disk drive, the disks rotate and an access arm, which contains two read/write heads, is used to read or write data from each surface of the disk.

Data Organization on a disk

- The sector method

To visualize the cylinder form of organization, imagine that a cylinder such as a tin can were dropped straight down through all the disks in the disk pack. Organization is vertical and access time is minimized. In this method the access arm has equal access to track 000 of all surfaces. All vertical tracks of a certain cylinder on a disk pack are lined up one beneath the other.

Disk Access to data

The primary factors that determine the time needed to access data are:

1. Seek time

The time it takes the read write head to get into position over a particular track. On an average, it could be 25 milliseconds. It depends upon the initial position of the read write head and the track number on which data is stored. In some hard disk drives each track has a separate read write head and then in that case the seek time is reduced to zero. This gives a very fast access to disk data.

2. Rotational Delay or Latency Time

Basically data is stored on sectors and it is read from and written on sectors. The time during which the read/write head waits in position until the record on the sector moves under it is called as Rotational time or latency time.

3. Data Transfer

The total time for data transfer is Seek Time +Rotational Time. The transfer of data between memory and disk. For example the data transfer rate for the IBM 3350 is 1,198,000 bytes per second.

2 Number Systems

2.1 Introduction

Computers use the two-state system called the *binary system*. The binary system has a base of 2. Each digit 0 or 1 of the binary system is called a bit which is the basic unit for storing data. Zero means *off* and 1 means *on*. Collection of 8 bits is a byte.1024 bytes is one kilo byte i.e. K. One million bytes is called one mega- byte i.e. MB and billions of bytes is expressed in terms of gigabytes (abbreviated GB).

A **word** is defined as the number of bits that constitute a common unit of data as defined by the computer system.

The Value of Each Digit in a Number is identified by:
1. The Digit
2. Position
3. Base (Radix)

Number System	Base	Symbols Used
Decimal	10	0,1,2,3,4,5,6,7,8, 9
Binary	2	0,1
Octal	8	0,1,2,3,4,5,6,7
Hexadecimal	16	0 – 9

2.2 Conversion

Conversion to decimal

$(148)_{10} = 1 \text{ x } 10^2 + 4\text{x}10^1 + 8\text{x}10^0 = 100+40+8 = 148$

$(10010100)_2 = 1\text{x}2^7 + 1\text{x}2^4 + 1\text{x}2^2 = 128+16+4 = (148)_{10}$

$(224)_8 = 2\text{x}8^2 + 2\text{x}8^1 + 4\text{x}8^0 = 128+16+4 = (148)_{10}$

$(94)_{16} = 9\text{x}16^1 + 4\text{x}16^0 = 144 + 4 = (148)_{10}$

Binary to Octal

- Divide the binary digit into groups of three from right
- Convert each group into one Octal digit

$(10010100)_2 = (\underline{010}\ \underline{010}\ \underline{100})_2 = (224)_8$

Binary to Hexadecimal

$(10010100)_2 = (\underline{1001}\ \underline{0100})_2 = (94)_{16}$

Hexadecimal to Binary

Convert the decimal equivalent of each Hexadecimal digit to four binary digit and combine.

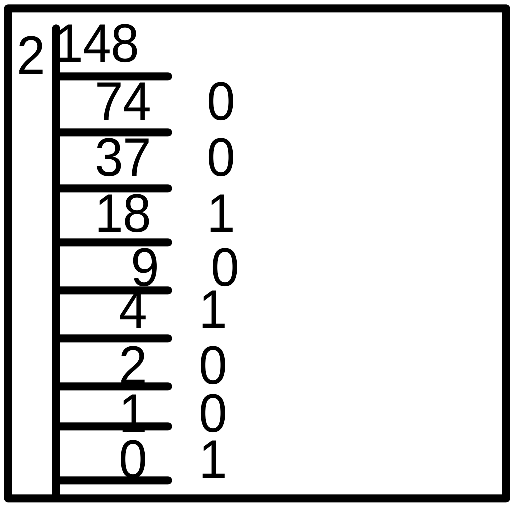
Consider $(A3)_{16}$

$(A)_{16} \equiv (10)_{10} \equiv (1010)_2$

$(3)_{16} \equiv (3)_{10} \equiv (0011)_2$

$(A3)_{16} \equiv (10100011)_2$

Decimal to Binary



Result : (10010100)2

2.3 Coding Schemes

The commonly used coding schemes for representing numbers, letters, and special characters are:

- EBCDIC
- ASCII.
- BCD

EBCDIC stands for **Extended Binary Coded Decimal Interchange Code**. It was established by IBM and used in IBM mainframe computers and uses 8 bits to represent a single character. EBCDIC (Extended Binary Coded Decimal Interchange Code) is a character encoding set used by IBM mainframes. Unlike virtually every computer system in the world which uses a variant of ASCII, IBM mainframes and midrange systems such as the AS/400 tend to use a wholly incompatible character set primarily designed for ease of use on punched cards.

EBCDIC uses the full 8 bits available to it. Also, EBCDIC has a wider range of control characters than ASCII

EBCDIC is the most common alternate character code but there are others.

ASCII stands for **American Standard Code for Information Interchange** and uses 7 bits for representing characters. The ASCII representation has been adopted as a worldwide standard by all.

BCD stands for **Binary Coded Decimal.** In this scheme each digit in the decimal representation is converted into its equivalent binary form

3 Computer Software

3.1 Introduction

Computer Software means programs, packages or lists of instructions, stored in storage media like floppies, hard disk, magnetic tape, etc. from where the same must be loaded in computer's primary memory (i.e. RAM) for getting used. There are 2 main categories:

- Systems Software
- Application Software

3.2 Systems Software

Supplier of computer normally along with the computer hardware, to act as link between the computer system & end user provides this. This user is tuned to the hardware configuration or set up & is an integral part of the computer system, with the help of which the user can optimize use of the computer resources to do maximum work in minimum time. This software includes (a) Operating System (b) Translators i.e. Assemblers, Interpreter & Compilers & (c) Linkers and Loaders. However, some computer professionals opine that DBMS (Data Base Management Systems) is also part of systems software, but rather it is application area software for managing the database.

3.2.1Translators

Compiler

It is the program that converts all the High Level Language instructions into Machine Language at a time. It translates the program instructions before the program is run. A compiled program does not need the presence of the source code while it is executing. A compiled program runs much faster than an interpreted program. Examples of programming languages that are compiled before execution include **C** and **COBOL**

Interpreter

It is the program that converts High Level Language into Machine Language one instruction at a time. It translates the program instructions at **run time.** A very popular programming language that is interpreted is the **BASIC**

3.2.2 <u>Operating System</u>

It is part of systems software & is a set of programs that control operations of the computer. i.e. the operations of the input, output and storage devices. It is the means of communication between the user and the computer even providing interactive mode between the computer and its user. It organizes the storage media, manages loading of software to RAM and establishes a link between the hardware and the user. Note that operating systems are also designed in such a manner as can be suitable on certain types of computers only & even the application packages or programs designed to run on particular operating system may not run on other operating system. Examples of the operating systems are:

- Batch Processing

In applications requiring periodical processing and not requiring on-line processing as soon as transaction takes place, the transactions can be batched suitably like payroll Batch, Inventory Batch, Accounts Batch, etc. and then the batches processed sequentially one after another, at end of certain period say, a month. In short master files in that case will get updated at end of each month. This type of processing started with second-generation computers.

- Real Time or on-line processing

In this type of processing the transactions are processed as soon as they are incurred, i.e. the accumulation of transactions of any period like the case of batch processing is not done. Here the master files get updated with inputting of every new transaction, which just took place. Railway or Airway Seat Reservations are best examples of areas requiring on-line processing. In such a system of processing the records must be updated with every new occurrence of transaction, so if there is breakdown of computer system there would be problem, thus back-up system must be efficient enough too, which makes this processing costly. This processing also started in second-generation computers. In short this processing is done in the interactive mode only.

- Multiprocessing

This system uses 2 or more CPUs connected to main computer and the CPUs communicate with each other through the main computer. The difference between distributed processing and this system is that no job is divided in this system but the different terminals using different CPUs independently handle more than one job.

- Time Sharing

It is interactive form of processing by several terminal users in communication with same computer at same time. Each terminal user is given fixed short period of time called "*time slot or time slice*" by Supervisor Program & Transfer Monitor to interact with main computer, so that no user has to wait in queue like that in multiprogramming.

3.2.3MS-DOS Commands

It is a set of programs, which interprets the commands given to the computer. When an application program is being executed it takes the help of the operating system for input/output operation.

Basic DOS commands

- DIR Command

The directory command displays the files you have stored on a disk.

- DIR/P is used to display the list page-wise.
- DIR/W is used to see only the filename with the extension (width-wise)
- DIR .dbf to see all files having extension of 'dbf'..

The extension of a file is usually a three-lettered word, which follows after the filename and a period. '*' is called a wild card. It means, "replace with any character". Another wildcard character is '?'. It means, "replace with one character".

- TYPE command

To check the contents of the file in the current directory.

- C:\>TYPE myfile

- DEL command

To delete one or more files in the current directory,

- DEL Bayer.* deletes all the files with filename Bayer and having any extension

3.3 Application Software

These are programs, which can do specific jobs only. They are developed by the user tailor-made to suit his requirements or they may be ready made packages developed by software developing companies for rather common activities like payroll, financial accounting, word processing, data processing, inventory control, etc. They are more economical compared to self-developed packages as their development costs are borne by several purchasers. But they may not be always be suited to needs of any particular organization in which case the software company which developed the package or its authorized representative can modify the software as to suit the client's requirements. Many organizations prefer to purchase ready made packages available in market because of economy, the package being tested & tried and as time lost in developing one's own package (particularly in debugging one's own software for errors that might have crept in while developing the package) is saved. However, the disadvantage of ready-made packages is that they may not always suit organization's requirement in all respects.

3.4 Programming Languages

3.4.1Levels of Language

Low Level Languages

These are closer to the language understood by the computer viz. 'machine language' that is made up of the symbols 0 and 1 only. Examples of low-level languages are:

- **Machine language** which is the lowest form of computer language.

- **Assembly language**, which is one level higher than machine language. This language uses abbreviations or mnemonic codes to replace numbers; A for

Add, C for Compare and so on. A translator is required to convert assembly code into machine code. Like machine language, the programs are difficult to read and understand

High-level languages

Written in an English-like manner. A translator, called a compiler is needed to convert high-level language into machine language. Examples are COBOL, Fortran, Basic, C, Pascal, Ada etc. Also called procedural languages where the computer is told how a task is done.

Fourth Generation Languages

Also known as Very High-Level Language. They are non-procedural in nature. Here, the users tell the computer what they want it to do and not how it is to be done. This leads to productivity, a key characteristic of fourth-generation languages. Examples are SQL (Structured Query Language).

A 4GL request looks something like this:

```
SUM UNITS BY MONTH BY CUSTOMER BY PRODUCT ON CUSTOMER
SUBTOTAL PAGEBREAK
```

The downside of 4GL's

4GL's are all not peaches and cream, since extra costs are involved and a large number of users tend to use such systems thereby making it more prone to crashes.

4 Flowcharts

Structured Programming technique

Structured programming techniques imply that the program development and coding must follow a top-down approach and structured programming constructs. Top-down approach implies planning the program as a hierarchy of modules.

Programming begins with top (main) module. When the main module is programmed second level modules are then programmed.

In order to support the discipline of top-down programming the following basic structured programming constructs are followed in programming:

Sequence: Sequence indicates that if there is no intervening control structure the flow of control normally passes from one instruction to another in sequence.

Selection: Selection indicates a decision. You may have one way, two way or multiway selection that is to say that you may branch out to one, two or multiple locations within the program depending on the decision.

Repetition: If a statement has to be executed more than once then the technique of repetition is applied. It saves the trouble of writing the same statement again and again.

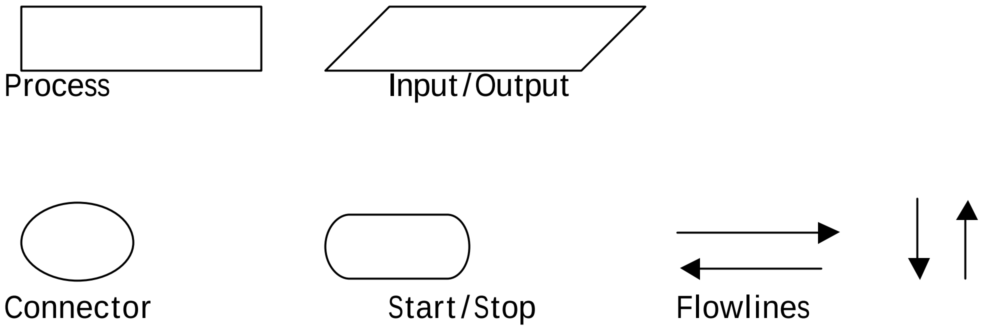
Program Flowchart

A program flowchart is a pictorial representation of logic required to accomplish a task .It includes all necessary steps to be included in a program. It is called flowchart, because it charts the flow of a program. It is a symbolic representation of each input, output and processing steps. Besides being a good method of writing down the algorithm, it is also a part of program documentation and helps in understanding, debugging and maintaining of programs.

Program Tools

In the flowcharting technique operations are represented by drawing appropriate symbol for an action. These flowchart symbols are connected by arrows to illustrate the sequence of operations.

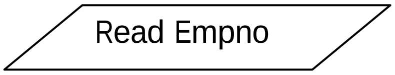
The basic symbols used in a program flowchart are as follow:



The Input/Output symbol

This symbol is used to represent the input/output operations (I/O operations) such as read and write.

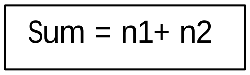
Eg.



The Process Symbol

This symbol is used to represent processes like assigning a value to a variable or adding a number. It has one arrow going inside to denote the input data to the process and another arrow leaving it to denote the processed data leaving this process.

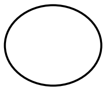
Eg.



The Terminal Symbol

This symbol indicates the beginning or end of a flowchart.

The Connector Symbol

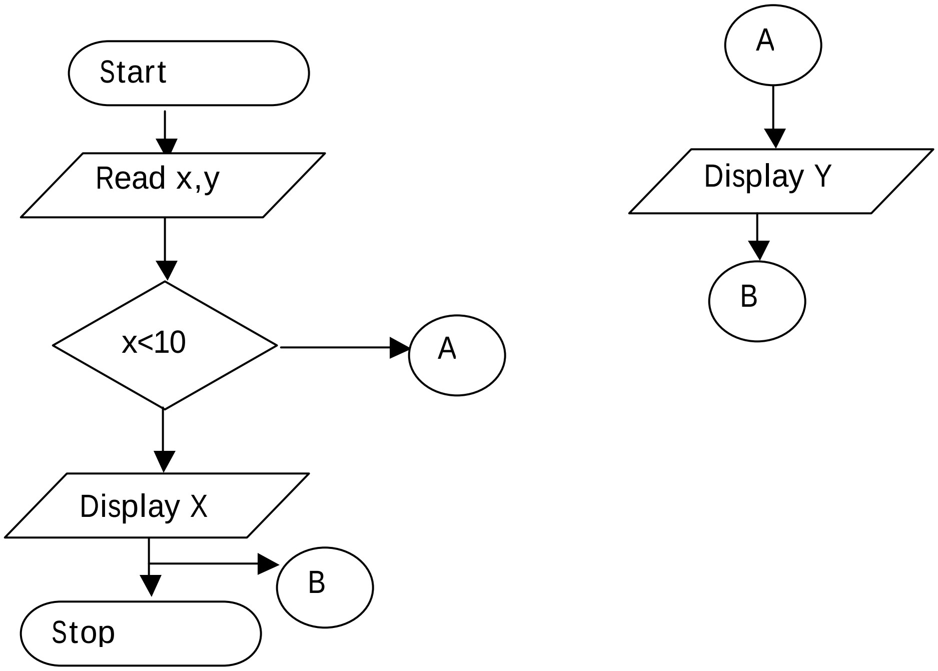


The connector symbol is used to maintain links when the flowchart runs longer then one page or the same diagram is continued at different locations of the same page.

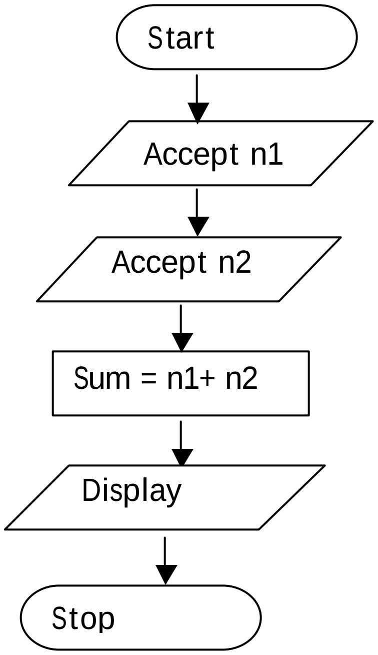
The Flowline Symbol

This symbol represents the direction of the flow of data flow in the flowchart. These are straight lines with arrowheads.

Eg of connector symbol.



Example 4-1 Flowchart to add two numbers



4.1 Algorithms checking methods

4.1.1Dry Run

Dry Run is a manual way of testing for algorithms for its correctness and functionality. In this method a table is prepared with columns for each variable used in the algorithm and the value of each variable is updated in the table as we proceed through the algorithm one step at a time.

Given below is a new example where a pseudocode is given and a Dry Run Table is made for the it. Example 1

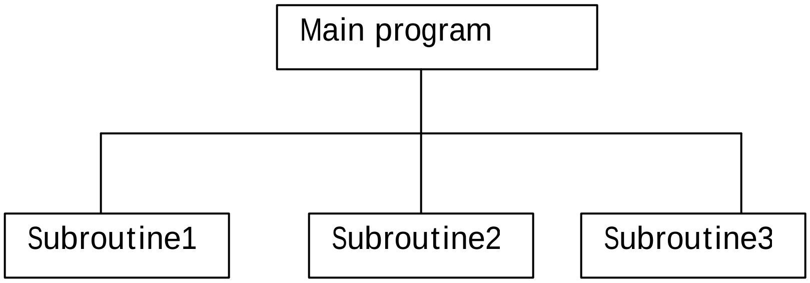
```
Start
X=5
Y=10
D=0
D=X+Y+(X*Y)
Y=Y+6
D=D+Y
DISPLAY X,Y,D
END
```

	x	y	D
Initial Value	5	10	0
After step 5	5	10	65
After step 6	5	16	65
After step 7	5	16	81

Example 4-2 Dry Run Table

Subroutines

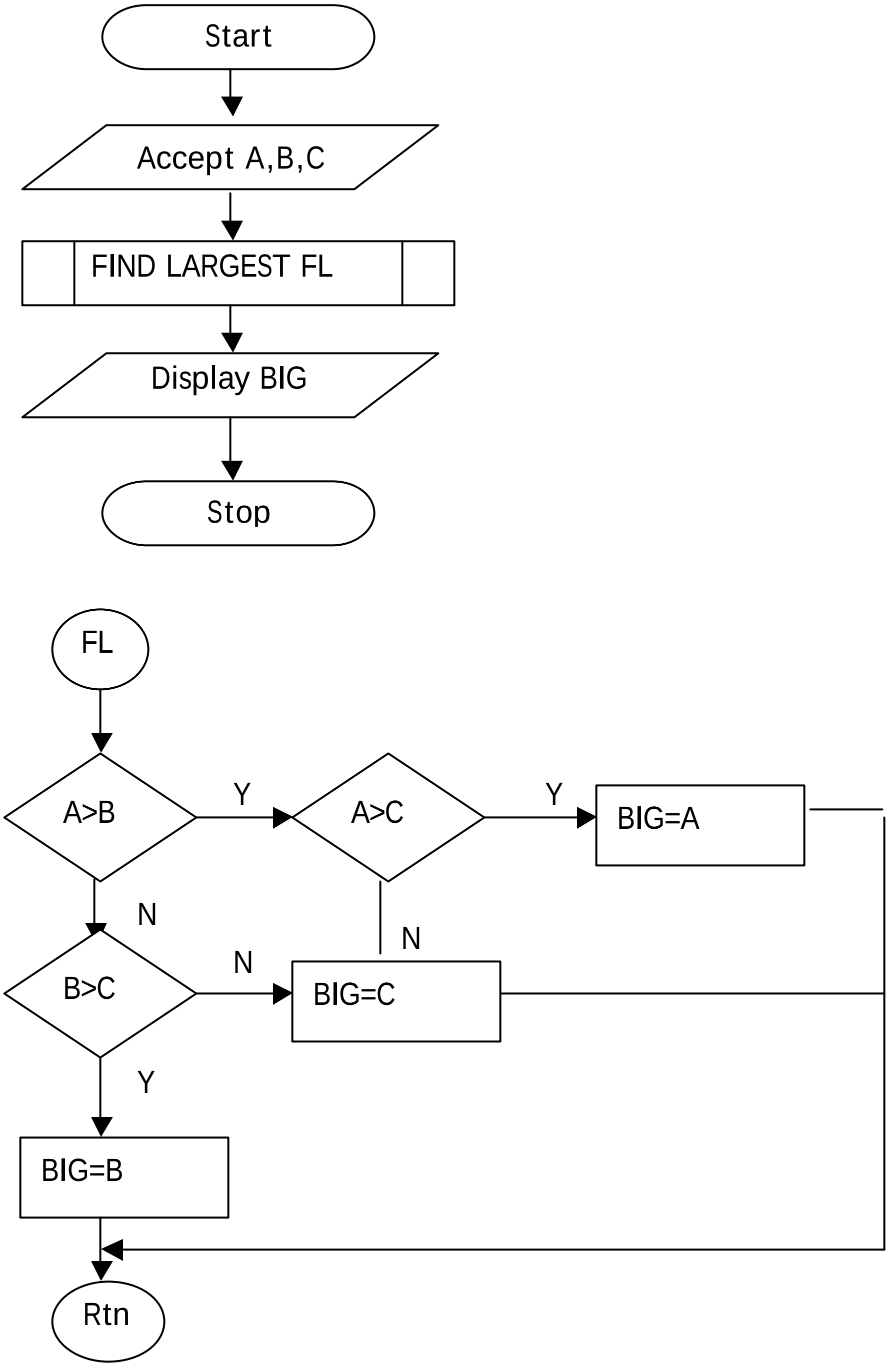
Subroutine, also called as subprogram is a small part in main (and the larger program). Generally, a major problem is subdivided into functionally different modules and for each module a separate program is written. Finally, once all modules are coded, they are integrated. so that it becomes one program. Each program is called subprogram or subroutine and each subroutine is functionally independent. This method of breaking a problem into a number of functionally independent module is called Top-Down decomposition. The following structure represents this concept.



The subroutine symbol

This symbol is filled with the name of the subroutine and the logic for this subroutine is drawn as a separate flowchart diagram.

Example1:



Here instead of given 'STOP' in subroutine we write' Return' which indicates that the control should be returned to the place, in the flowchart from where it branched and then the algorithm proceeds in the calling flowchart until it encounters a STOP instruction.

5 File Handling

Sequential File Organization: This is the simplest way of storing and retrieving the records in a file. Each record is stored one after the other and is retrieved in the same sequence. The first record is placed in the beginning second record is placed in the second position and so forth.

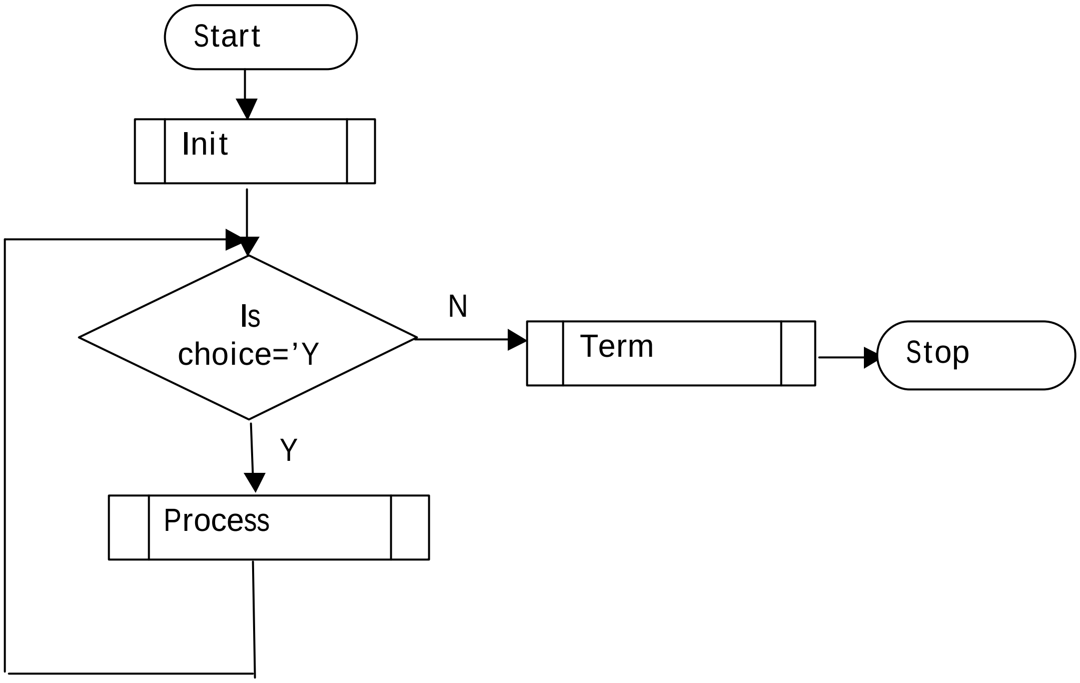
Reading sequential file: To read a record from the sequential file the search starts from the first record, proceeds through all records till it reaches the record in question. If the record to be processed happens to be the tenth record the system reads the first record then the second and so on until the tenth one is reached. It cannot directly go to the tenth record.

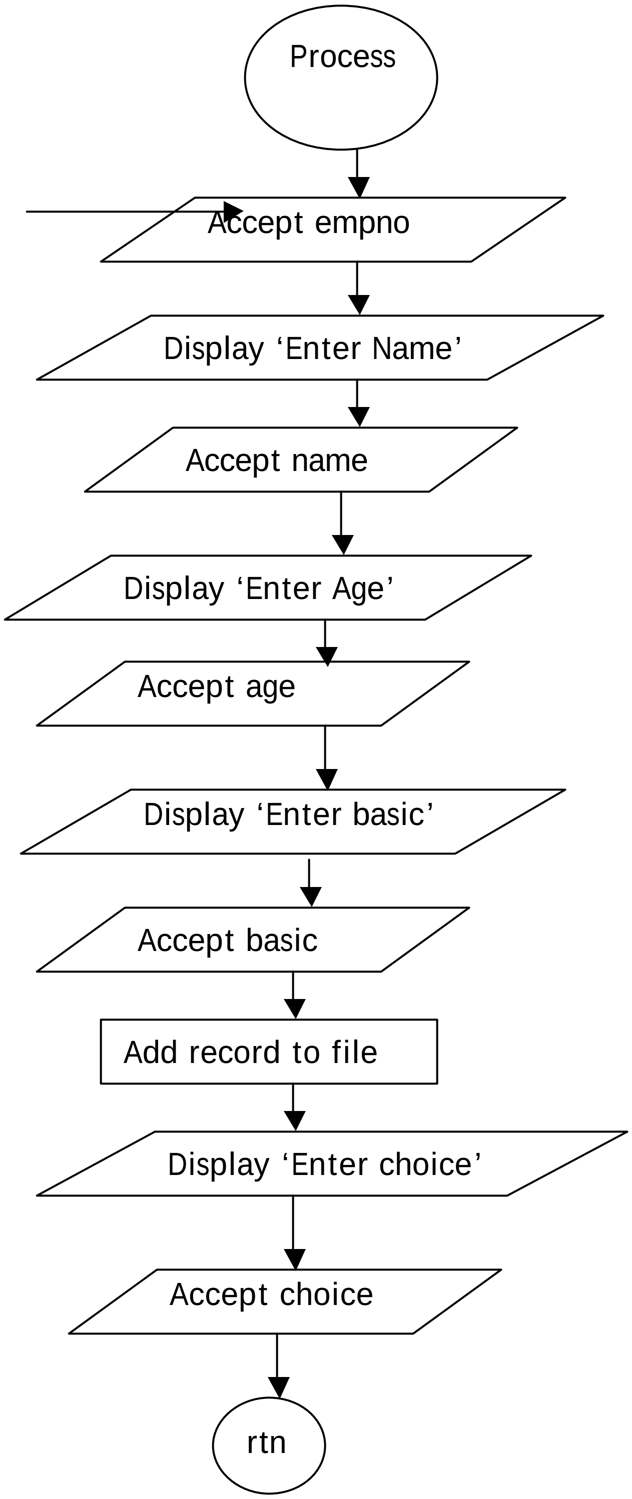
Data File

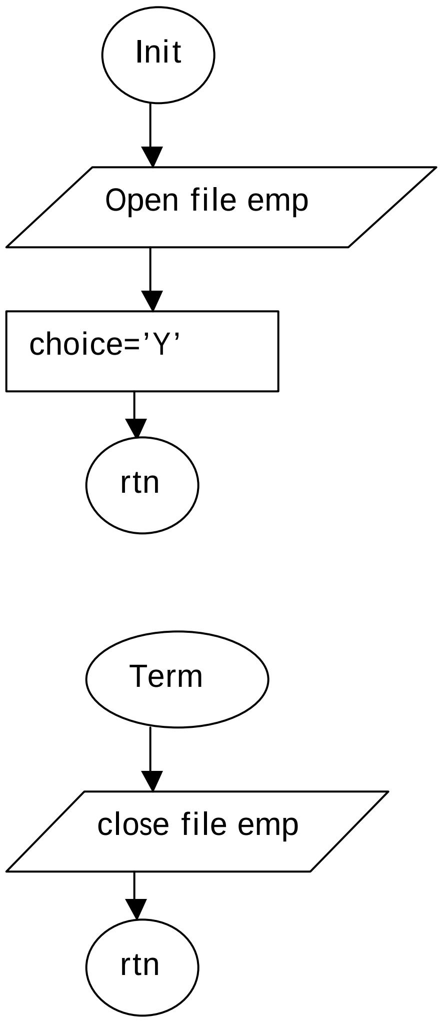
141	Roja	25	110	Rama	35	105	Roja	17	106	Uma	45

File with 4 records
Each record had 3 fields : 1. Employee No
2. Name
3. Age

Example 1: The flowchart creates a file with five fields, Employee No,Name,Age,Basic Salary and Allowance.




Process
Accept empno
Display 'Enter Name'
Accept name
Display 'Enter Age'
Accept age
Display 'Enter basic'
Accept basic
Add record to file
Display 'Enter choice'
Accept choice
rtn


Init
Open file emp
choice='Y'
rtn
Term
close file emp
rtn

6 Reports

Often the most important feature of an information system is the reports it produces. Reports are the disciplined way of presenting the data stored on the disk in a well-formatted manner so that one can take decisions based on it.

Report components
The report consists of a set of components, which are found in almost all the reports are:

Report title : Report title describes the information contained in the report. The report title should appear at the beginning of the report. If the report consists of several pages subsequent pages may also include page heading.

Detail section : Detail section in the report describes the contents of each record in a file. For example. in the invoice report the particulars of various items the quantity rates and cost are outlined in detail section.

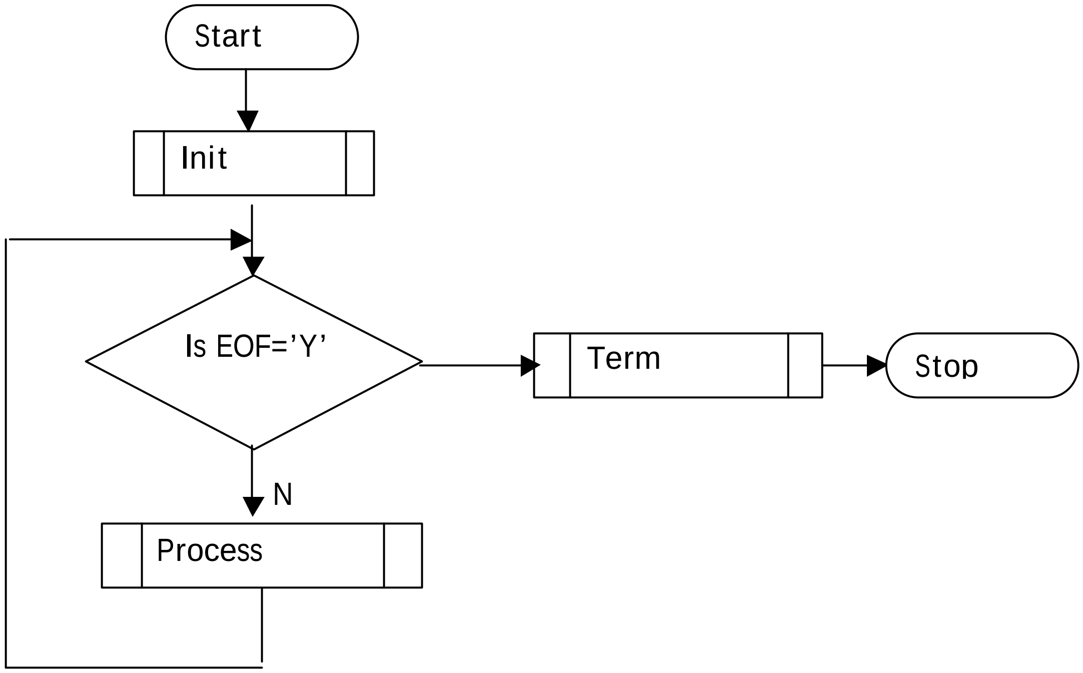
Summary section : Summarv section in the report presents the information in a brief form

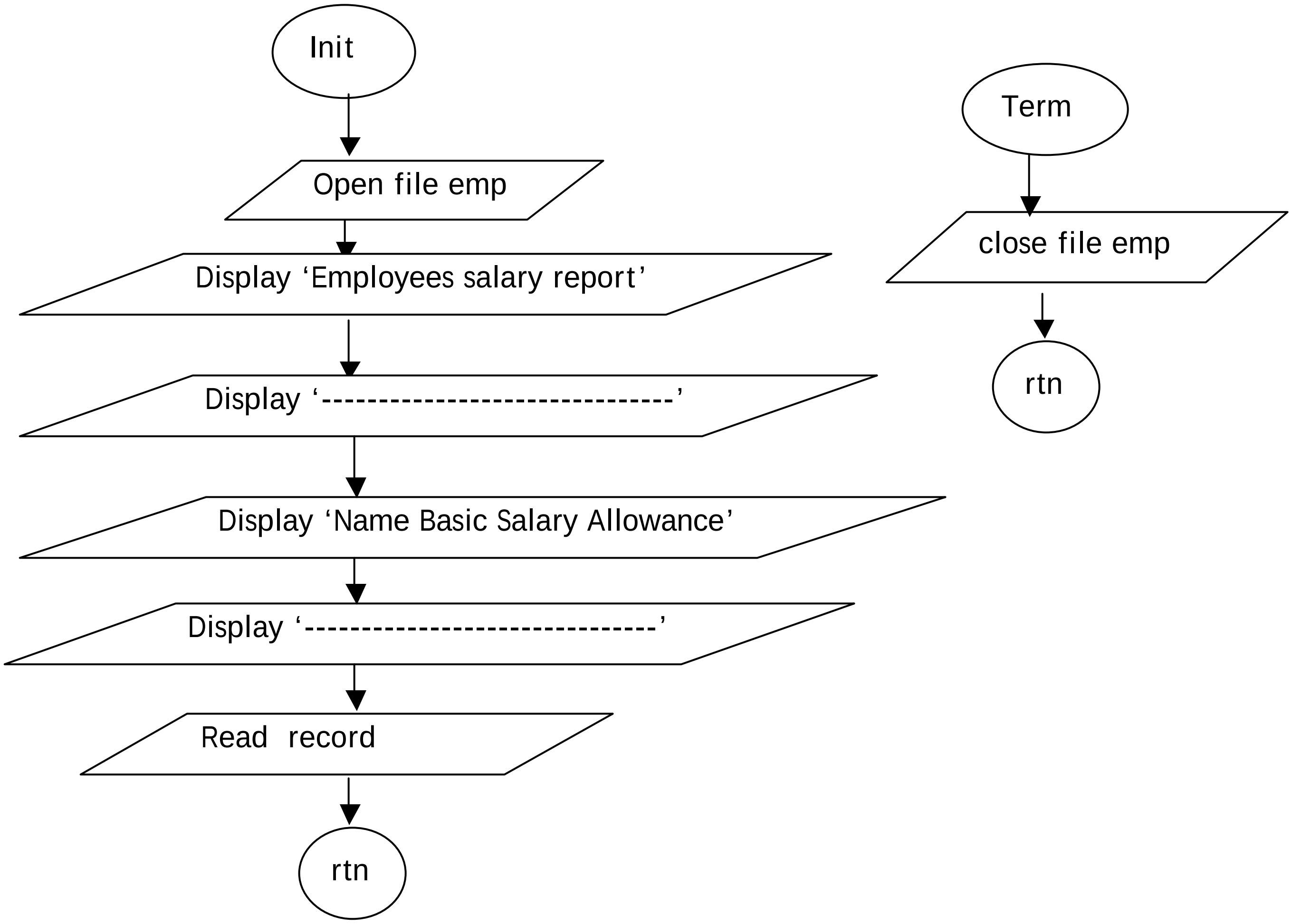
Page number : A sequential page number is usually included in multi-page reports.

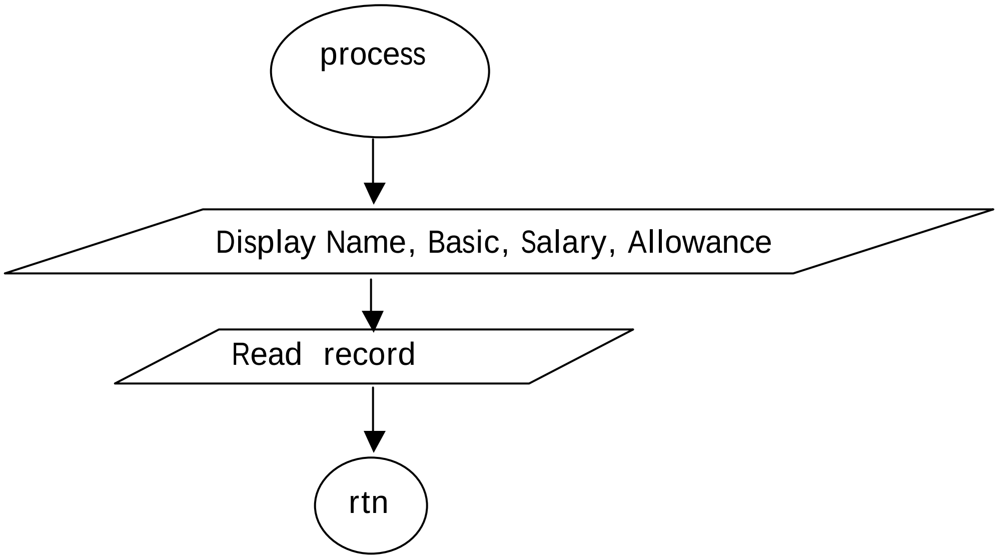
Report date: The report date specifies the date on which the report was generated.

Footer: Footer is used at the end of the page to print page no. etc.

Header : Header is used to print same heading like company name etc. to be printed on all pages of report.


Start
Init
Is EOF='Y'
Term
Stop
N
Process


Init
Open file emp
Display 'Employees salary report'
Display '-------------------------------'
Display 'Name Basic Salary Allowance'
Display '-------------------------------'
Read record
rtn
Term
close file emp
rtn


process
Display Name, Basic, Salary, Allowance
Read record
rtn

7 Control Break

Usually reports, which run into pages without giving any meaningful substance on group wise totals etc., make little sense. In many cases one would like to see group totals based on an application requirements. The technique through which such printing is handled is called control break technique. Control break is a matter of building the algorithm such that break in a report are generated in a controlled fashion and whenever a break occurs the algorithm goes on to perform certain activities usually it is printing a subtotal. This can be explained and understood better with an example.

Let us consider a file consisting of details of sales of different salesman *for* a month. Day to day sales are entered into the file. At the end of the month it is required to print the details of sales of every salesman along with the total sales *for* every salesman. Hence the file needs to either be sorted or indexed.

The fields of the Sales file are as follows:

SALESMANNO	CHARACTER	5
SALESNAME	CHARACTER	20
DT_OF_SALE	DATE	8
SALES	NUMERIC	5

The records of each salesman can be grouped by arranging them in the order of the salesman number. This can be done by sorting the file on the field SALESMANNO because the salesman number of each salesman is unique i.e. no two salesmen can have the same SALESMANNO. Once the records pertaining to each salesman is grouped together we can calculate the total sales for every salesman by adding up the sales of every record till the SALESMANNO changes. When the key field (the field on which the file is sorted) changes, a break is said to have occurred, and because this occurrence is controlled by design of the algorithm it is called a control break.

Sample Data

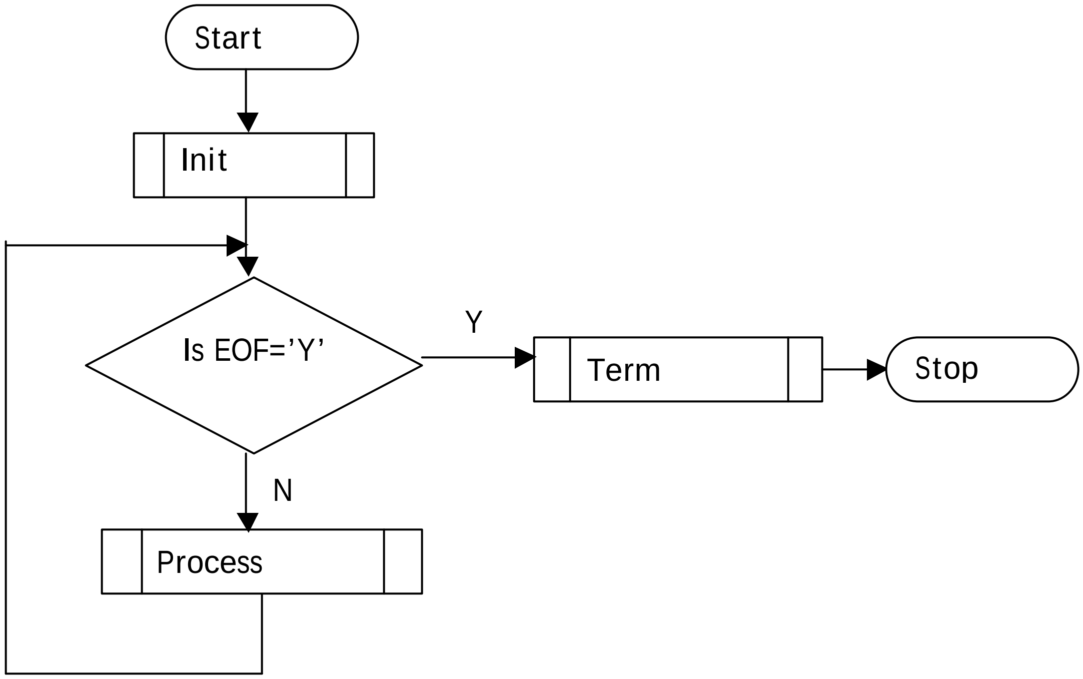
SALESMANNO	SALESMANNAME	DT_OF_SALE	SALES
E0001	UMA	01/02/93	6700
E0001	UMA	01/02/93	2800
E0002	KLIEN	01/09/93	8700
E0002	KLIEN	01/11/93	1200
W0001	TED	01/01/93	4500
W0001	TED	01/05/93	9800

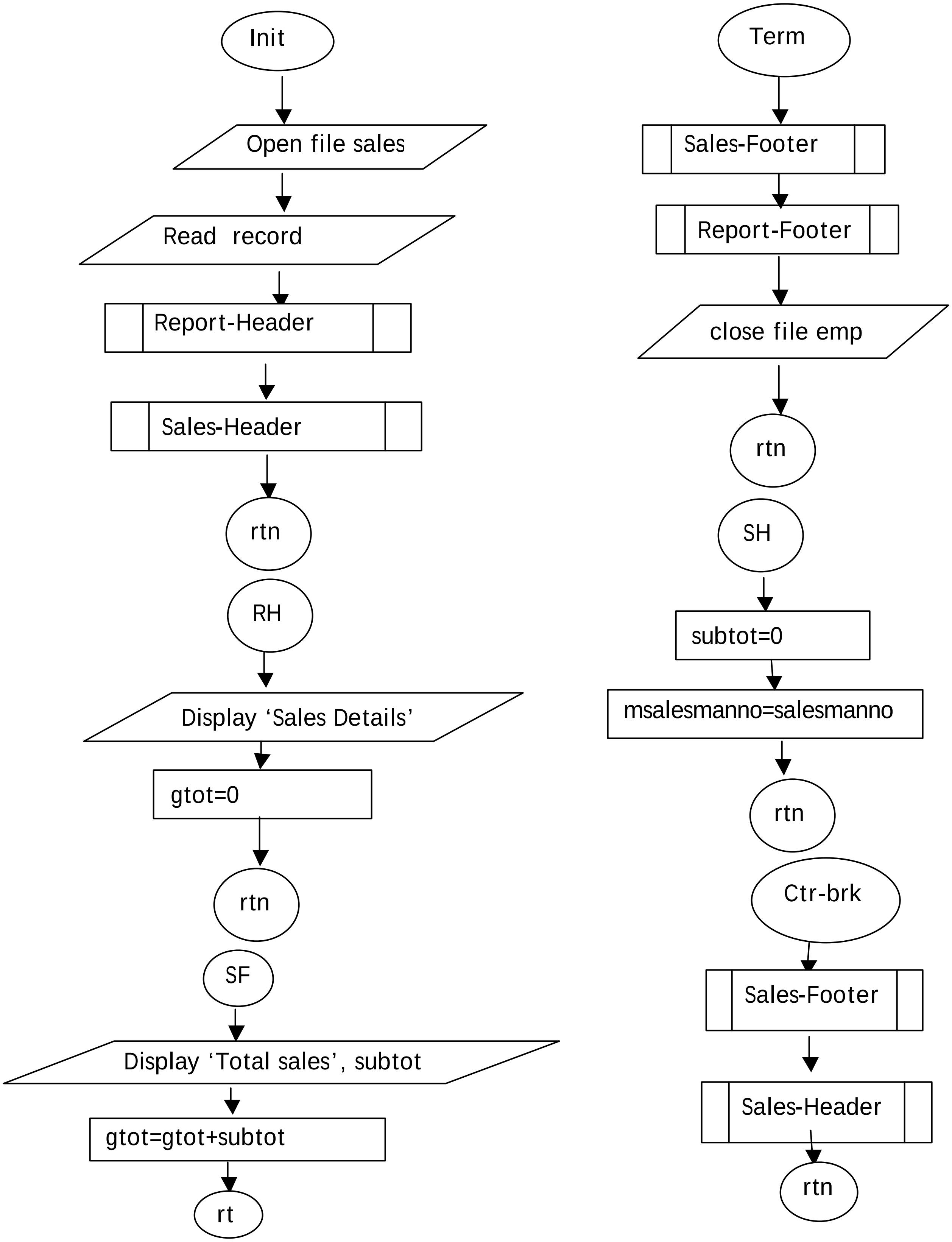
Sample Report

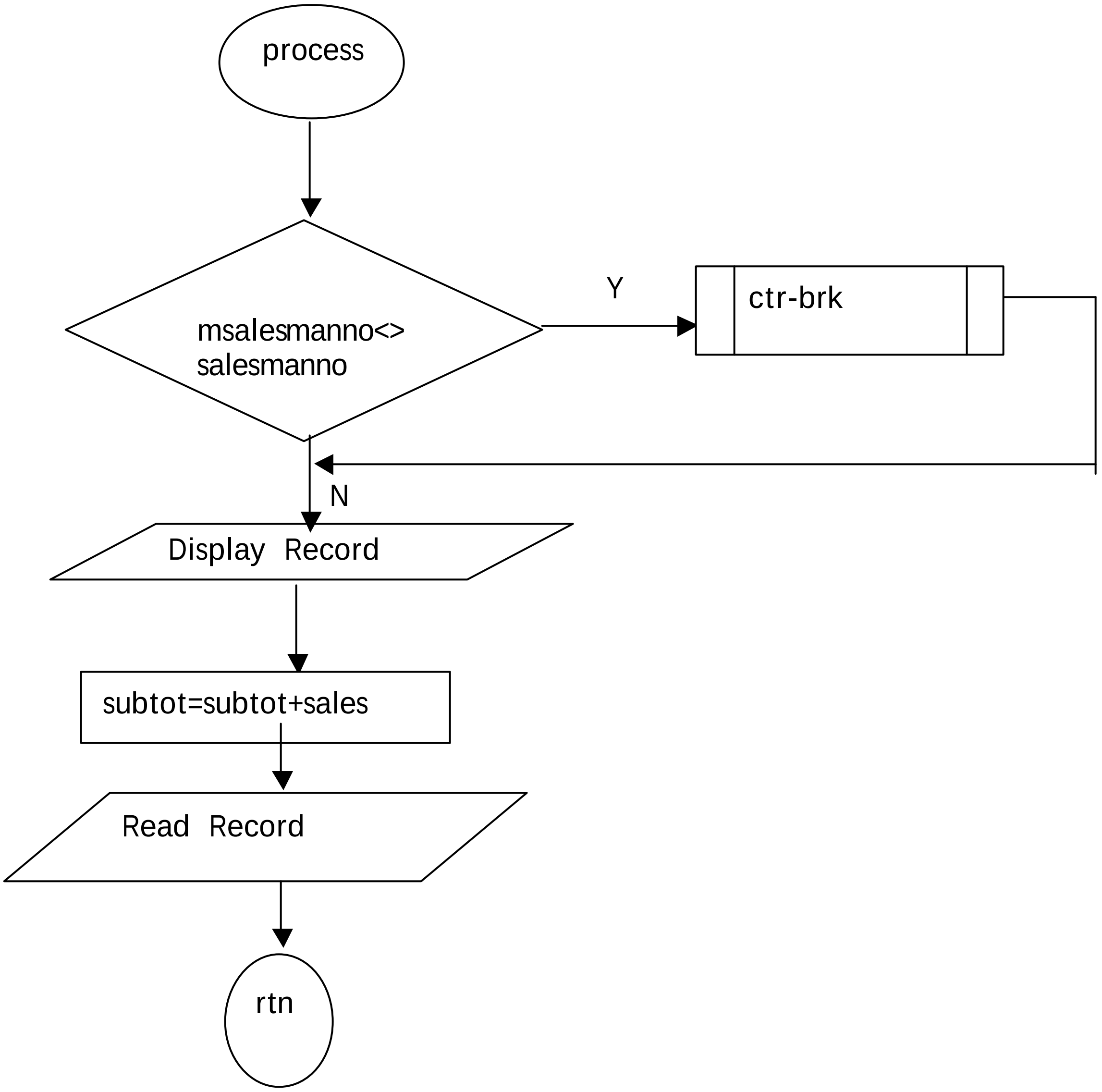
```
                              Sales Details

SALESMANNO – E0001

DATE OF SALE                          SALES AMOUNT
01/02/93                                     6700
01/11/93                                     2800
-------------------------------------------------------------------------------
TOTAL SALES BY SALESMAN E0001                9500
-------------------------------------------------------------------------------

SALESMANNO – E0002

DATE OF SALE                          SALES AMOUNT
01/09/93                                     8700
01/05/93                                     1200
-------------------------------------------------------------------------------
TOTAL SALES BY SALESMAN E0002                9900
-------------------------------------------------------------------------------

SALESMANNO – E0003

DATE OF SALE                          SALES AMOUNT
01/01/93                                     4500
01/05/93                                     9800
-------------------------------------------------------------------------------
TOTAL SALES BY SALESMAN E0003                14300
-------------------------------------------------------------------------------

TOTAL SALES                                  33700
```

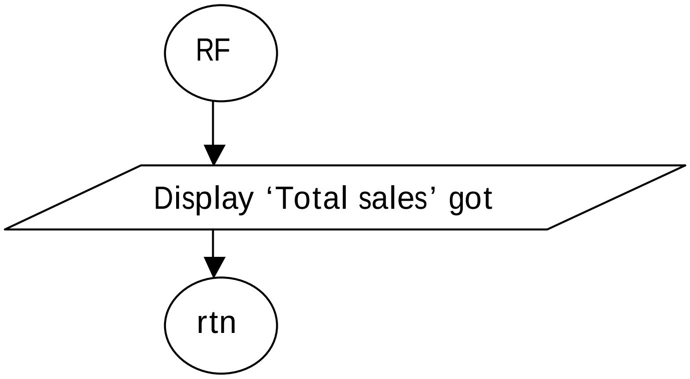
How to check for control break

The first thing that must be done in mind before going any further is that the file must be sorted on the field on which the grouping of records has to be done. After sorting the records on the field on which the records are to be grouped, store the value of the key field of the first record in a memory variable. Then compare the key field of the subsequent records with that memory variable as long as it matches. It means that you are printing records of the same group, when it does not match it means a control break has occurred. Then the algorithm must print the totals for the previous set of records. Store the new value of the key field in the memory variable and proceed.


Start
Init
Is EOF='Y'
Y
Term
Stop
N
Process


Init
Open file sales
Read record
Report-Header
Sales-Header
rtn
RH
Display 'Sales Details'
gtot=0
rtn
SF
Display 'Total sales', subtot
gtot=gtot+subtot
rt
Term
Sales-Footer
Report-Footer
close file emp
rtn
SH
subtot=0
msalesmanno=salesmanno
rtn
Ctr-brk
Sales-Footer
Sales-Header
rtn


process
msalesmanno<> salesmanno
Y
ctr-brk
N
Display Record
subtot=subtot+sales
Read Record
rtn


RF
Display 'Total sales' got
rtn

8 Database

A **database** is a collection of interrelated files stored together with minimum redundancy. A database management system (DBMS) is software that helps you organize data in a way that allows fast and easy access to the data. You can create, modify, store and retrieve data in a variety of ways.

Add database concepts: entity, attribute & relationship

Benefits are:

- Integrated Files

Separate files can be joined together

- Reduced Redundancy

In a database, data is usually stored in one place. So the amount of duplicate data is reduced. Updating can be done quickly and efficiently without having to track down the repeated data.

- Shared data

Different people can share data in the files.

- Centralized security

When data is all in one place, you have better control over access to it.

A structure of a database is dependent on data models. A data model is a collection of conceptual tools that describe the data, data relationships and consistency constraints. The models are:

- Relational
- Network
- Hierarchical

8.1.1Relational Database Model (RDM)

The ER model consists of a set of basic objects called **Entity** and the **Relationship** among these entities.

Users as tables perceive all the data. The structures are simple and easy to build. Data redundancy is controlled to a greater extent and faster access of data is possible. Allows many-to-many relationships.

Entity
It is an object that exists and is distinguishable from other objects. Ex: a person, a book etc.

Relationship
An association between two or more entities. Ex: customer places an order.

- Mapping Cardinalities

The ER model defines certain constraints to which data in the database must satisfy. These constraints are called Mapping Cardinalities. They are:

- One-to-One mapping
- One-to-many mapping
- Many-to-one mapping
- Many-to-Many mapping

One-to-one Relationship
An entity in A is associated with at most one entity in B and an entity in B is associated with at most one entity in A. Ex: Every trainee has one and only one sponsor and every sponsor sponsors one and only one trainee.

One-to-many Relationship
An entity in A is associated with any number of entities in B. An entity in B, however can be associated with at most one entity in A. Ex: Every employee works in exactly one dept. A dept staffs none, one or many employees.

Many-to-one Relationship
An entity in A is associated with at most one entity in B, and an entity in B is associated with any number of entities in B.

Many-to-many Relationship
A professional association may have none, one or more consultants and a consultant may belong to none, one or many associations.

The entities and the relationships in this model are represented using ER diagram:

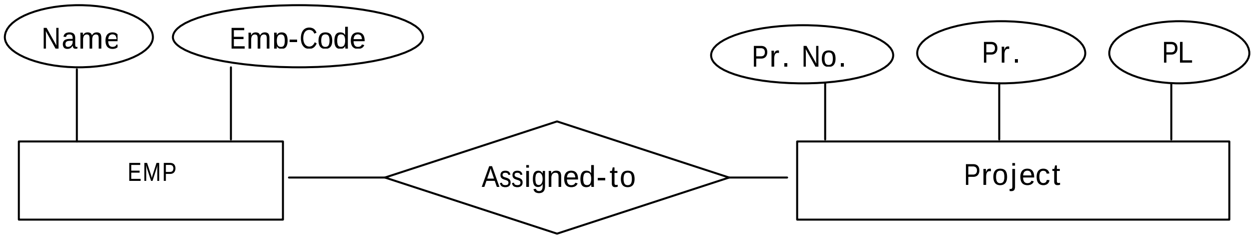


Figure 8-1 E-R Diagram

Emp_Code	Name
6550	Anjali
6551	Meena
6552	Sheetal
6553	Malti
6554	Vinita

Emp.Table

PR_No	PR_Name	PL
01	SBU	Malti
02	JBU	Vinita
03	MIS	Meena

Project

Emp_Code	PR_No
6553	01
6554	02
6551	03

Assign -To

8.1.2Data Abstraction

There are three levels:

- External level
- Conceptual level
- Internal level

The inter relationship among these three levels of abstraction is called the Architecture of Database System.

External level
That portion of the database structure that is processed by a particular application.

Conceptual level
The logical view of the database or the structure of the entire database.

9 Appendix - A

9.1 Assignments

Draw flowcharts for the following problems:

1. Display the name of the company.
2. Accept a name from the user. Display a greeting “Hello” followed by the name.
3. Accept 2 numbers from the user. Display the maximum no. If both the numbers are equal display the message “Equal”.
4. Display the series 1,2,3…25.
5. Modify the above problem the accept the start and end value from the user. Ensure that the end value is greater than the start value.
6. Display the maximum & minimum of the numbers entered by the user. Accept numbers, till the user enters a 0 (zero).
7. Display the multiplication tables for a given no.
8. Calculate the payable amount. You are given invoice no, customer code, invoice amount. Discount is 3% of the amount for amounts exceeding 5000. Tax is 1% of the amount except for customer codes ‘X’, ‘P’
9. Accept the following details from the user

   Empcode (2001 onwards, should be generated)
   Name
   Department (Sales, Purchase, Admin, S/W)
   Designation (Programmer, Analyst, Consultant for S/W department)
   (Jr. Officer, Sr. Officer, Manager for other departments)
   Salary (Should be between 2000 to 20000).

   Store these details in a file. Continue accepting details till the user enters ‘N’.

10. Display the details of all employees in department ‘Sales’. At the end display the average salary for the department
11. Display the details of all employee in the following format
    Note : File is sorted in the order of department

XYZ Co. Ltd

Department :

Sr. No. Emp code Name Salary

Department Total :

Department :

Sr. No. Emp code Name Salary

Department Total :

Grand Total :

12. Display the details of all employee in the following format
    Note : File is sorted in the order of department, designation

XYZ Co. Ltd

Department :

Designation :

Sr. No. Emp code Name Salary

Designation Total :

Designation :

Sr. No. Emp code Name Salary

Designation Total :

Department Total :

Department :			
Designation :			
Sr. No.	Emp code	Name	Salary
	Designation Total	:	
Designation :			
Sr. No.	Emp code	Name	Salary
	Designation Total	:	
	Department Total	:	
	Grand Total	:	

10 Appendix - B

10.1 Evaluation Procedures

The Evaluation of participants' would be based on the Tests and Assignments conducted.

A participant who obtains less than 50% marks in test or less than 50% in assignments will be declared as failed.

60% weightage is given to assignments and 40% to test.

The grades as per the weighted marks are as follows

94% and above	A+
80% – 93%	A
60%-79%	B
50%-59%	C
Below 50%	F

11 Appendix - C

11.1 Bibliography / References

1. V. Rajaraman, Fundamentals of Computers, Prentice Hall of India, New Delhi

2. H.L. Capron, Computers – Tools for an Information Age, The Benjamin/Cummings Publishing Company, Inc, California

3. Graham C Lester, Data Processing Vol.1 – Hardware and Programming, Pitman, London

4. Hunt and Shelly, Fundamentals of Computers

12 Appendix - D

12.1 Table of Contents – Figures

12.2 Table of Contents – Examples