Chapter 09

Input-Output Devices

Computer Fundamentals - Pradeep K. Sinha & Priti Sinha

Learning Objectives

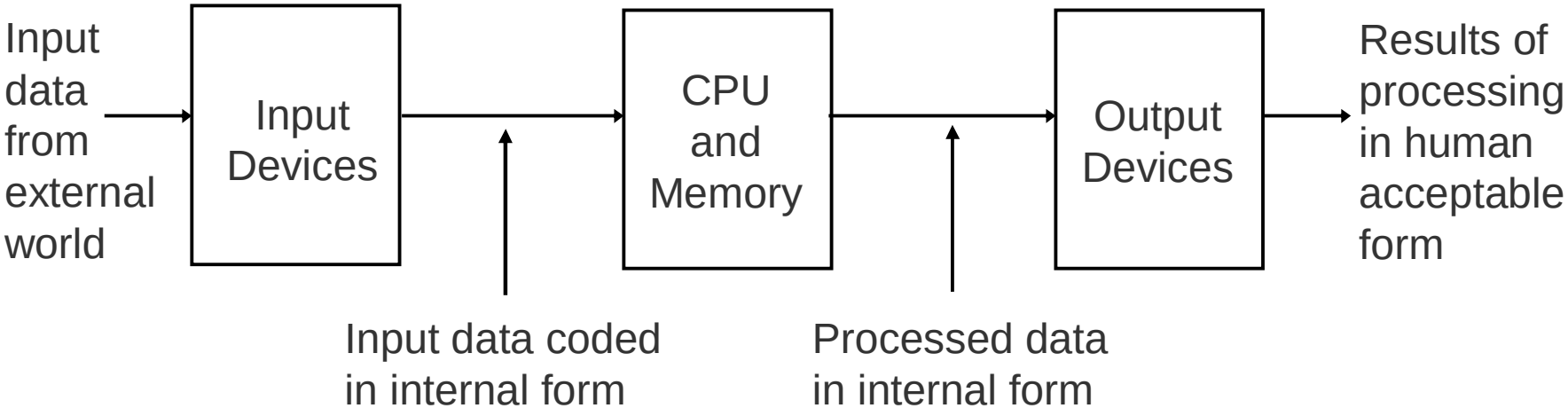
In this chapter you will learn about:

- Input/Output (I/O) devices
- Commonly used input devices
- Commonly used output devices
- Other concepts related to I/O devices

I/O Devices

- Provide means of communication between a computer and outer world
- Also known as peripheral devices because they surround the CPU and memory of a computer system
- Input devices are used to enter data from the outside world into primary storage
- Output devices supply results of processing from primary storage to users

Role of I/O Devices



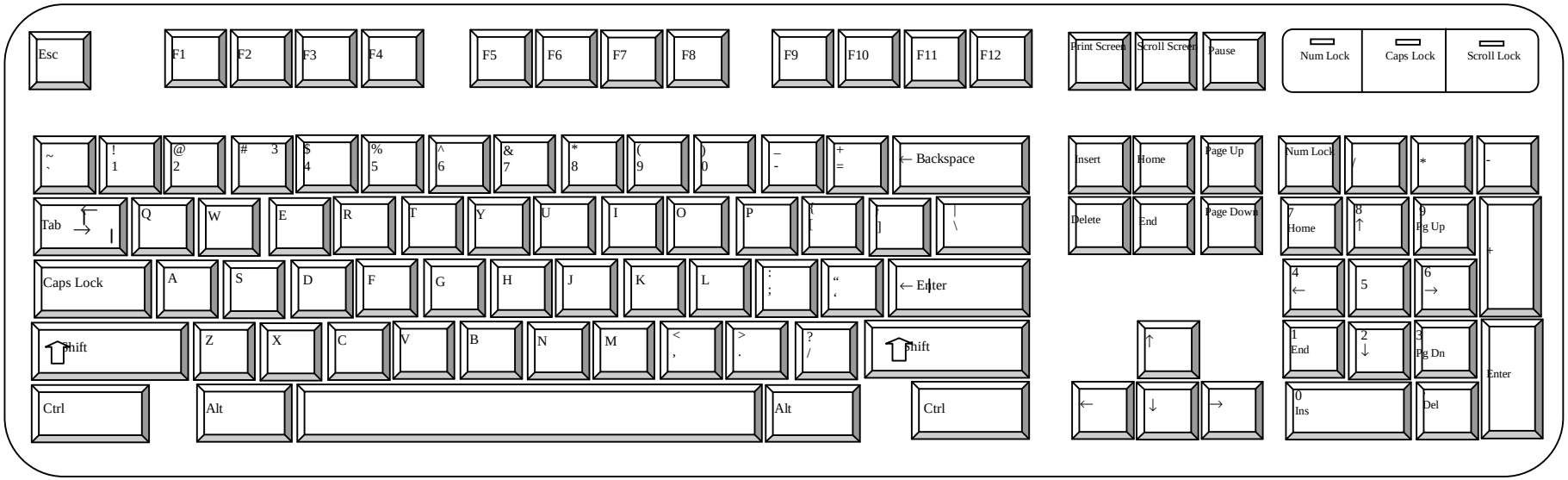
Commonly Used Input Devices

- Keyboard devices
- Point-and-draw devices
- Data scanning devices
- Digitizer
- Electronic cards based devices
- Speech recognition devices
- Vision based devices

Keyboard Devices

- Allow data entry into a computer system by pressing a set of keys (labeled buttons) neatly mounted on a keyboard connected to a computer system
- 101-keys QWERTY keyboard is most popular

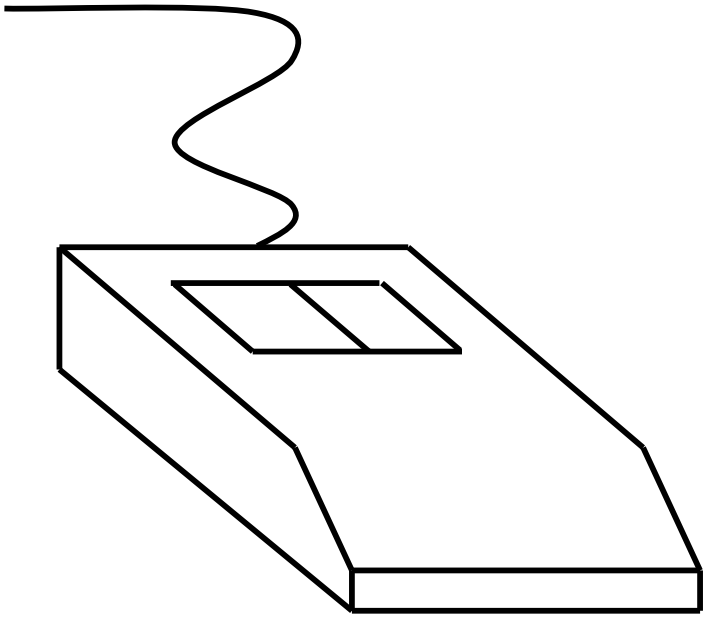
The Layout of Keys on a QWERTY Keyboard



Point-and-Draw Devices

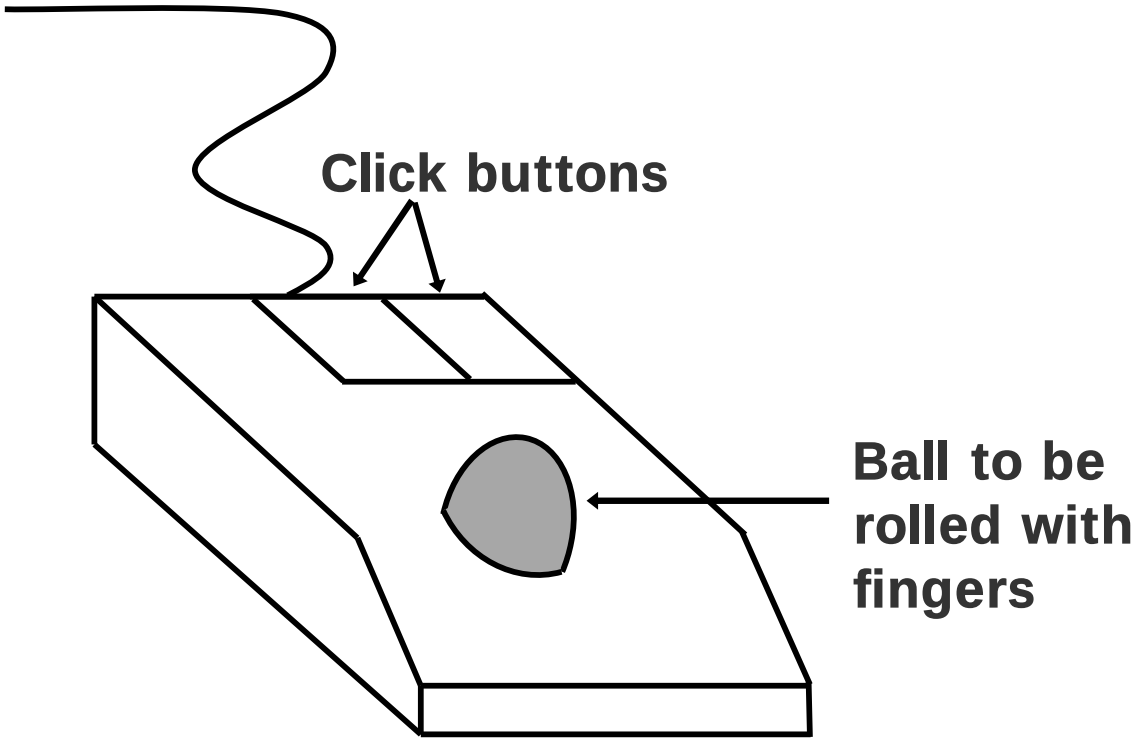
- Used to rapidly point to and select a graphic icon or menu item from multiple options displayed on the *Graphical User Interface (GUI)* of a screen
- Used to create graphic elements on the screen such as lines, curves, and freehand shapes
- Some commonly used point-and-draw devices are mouse, track ball, joy stick, light pen, and touch screen

Mouse



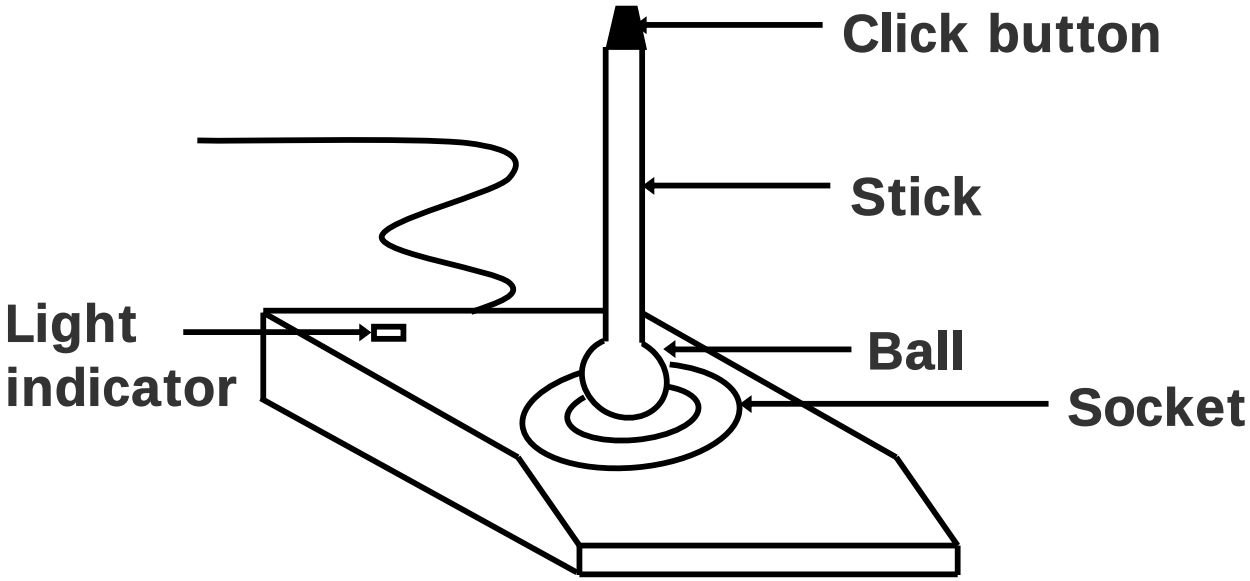
Commonly used in personal computers and workstations

Trackball



Commonly used in laptop (notebook) computers

Joystick



Commonly used for video games, flight simulators, training simulators, and for controlling industrial robots

Electronic Pen

- Pen-based point-and-draw device
- Used to directly point with it on the screen to select menu items or icons or directly draw graphics on the screen
- Can write with it on a special pad for direct input of written information to a system
- Pressure on tip of a side button is used to cause same action as *right-button-click* of a mouse

Touch Screen

- Most simple, intuitive, and easiest to learn of all input devices
- Enables users to choose from available options by simply touching with their finger the desired icon or menu item displayed on the screen
- Most preferred human-computer interface used in *information kiosks* (unattended interactive information systems such as automatic teller machine or ATM)

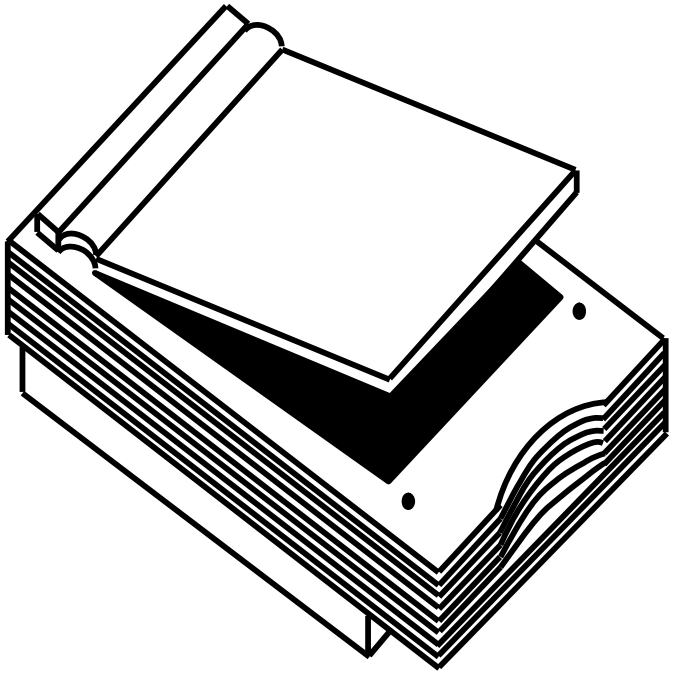
Data Scanning Devices

- Input devices that enable direct data entry into a computer system from source documents
- Eliminate the need to key in text data into the computer
- Due to reduced human effort in data entry, they improve data accuracy and also increase the timeliness of the information processed
- Demand high quality of input documents
- Some data scanning devices are also capable of recognizing marks or characters
- Form design and ink specification usually becomes more critical for accuracy

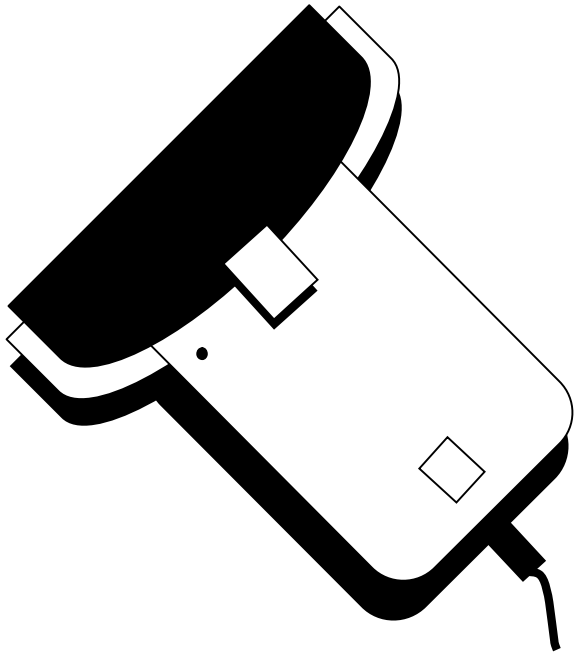
Image Scanner

- Input device that translates paper documents into an electronic format for storage in a computer
- Electronic format of a scanned image is its bit map representation
- Stored image can be altered or manipulated with an image-processing software

Two Common Types of Image Scanners



A flat-bed scanner



A hand-held scanner

Optical Character Recognition (OCR) Device

- Scanner equipped with a character recognition software (called OCR software) that converts the bit map images of characters to equivalent ASCII codes
- Enables word processing of input text and also requires less storage for storing the document as text rather than an image
- OCR software is extremely complex because it is difficult to make a computer recognize an unlimited number of typefaces and fonts
- Two standard OCR fonts are OCR-A (American standard) and OCR-B (European standard)

Optical Mark Reader (OMR)

- Scanner capable of recognizing a pre-specified type of mark by pencil or pen
- Very useful for grading tests with objective type questions, or for any input data that is of a choice or selection nature
- Technique used for recognition of marks involves focusing a light on the page being scanned and detecting the reflected light pattern from the marks

Sample Use of OMR

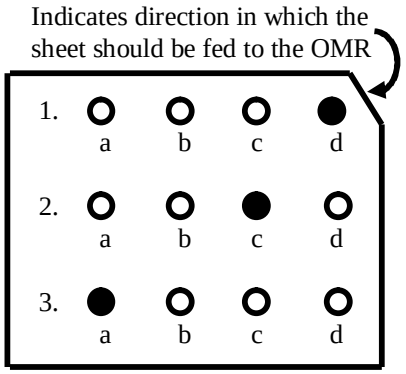
For each question, four options are given out of which only one is correct. Choose the correct option and mark your choice against the corresponding question number in the given answer sheet by darkening the corresponding circle with a lead pencil.

1. The binary equivalent of decimal 4 is:
- **a) 101**
- **b) 111**
- **c) 001**
- **d) 100**

2. The full form of CPU is:
- **a) Cursor Positioning Unit**
- **b) Central Power Unit**
- **c) Central Processing Unit**
- **d) None of the above**

3. Which is the largest unit of storage among the following:
- **a) Terabyte**
- **b) Kilobyte**
- **c) Megabyte**
- **d) Gigabyte**

(a) Question sheet



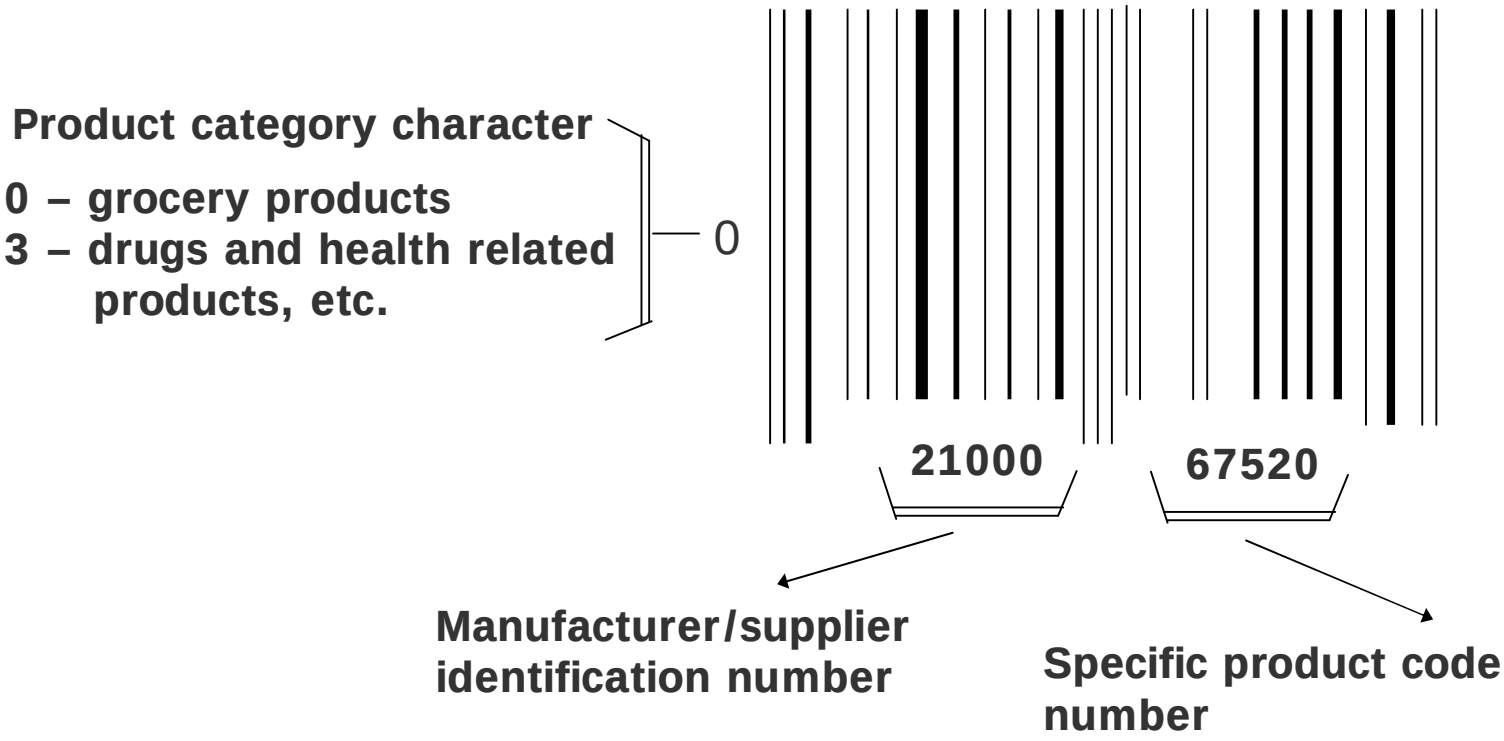
(b) Pre-printed answer sheet

A sample use of OMR for grading tests with objective type questions

Bar-code Reader

- Scanner used for reading (decoding) bar-coded data
- Bar codes represent alphanumeric data by a combination of adjacent vertical lines (bars) by varying their width and the spacing between them
- Scanner uses laser-beam to stroke across pattern of bar code. Different patterns of bars reflect the beam in different ways sensed by a light-sensitive detector
- Universal Product Code (UPC) is the most widely known bar coding system

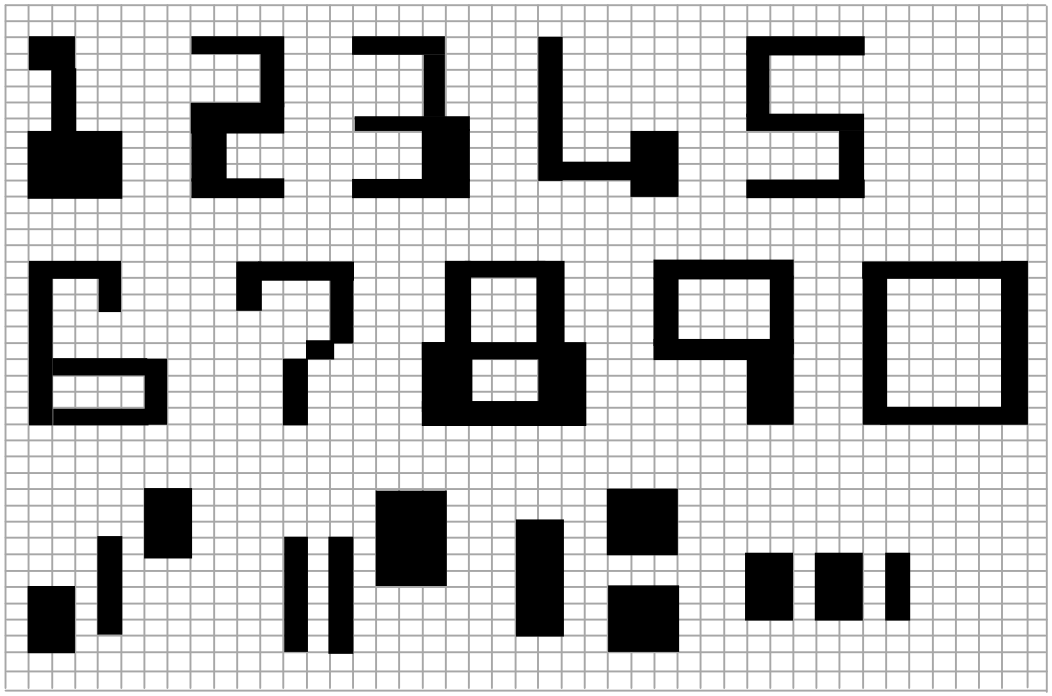
An Example of UPC Bar Code



Magnetic-Ink Character Recognition (MICR)

- MICR is used by banking industry for faster processing of large volume of cheques
- Bank's identification code (name, branch, etc.), account number and cheque number are pre-printed (encoded) using characters from a special character set on all cheques
- Special ink is used that contains magnetizable particles of iron oxide
- MICR reader-sorter reads data on cheques and sorts them for distribution to other banks or for further processing

MICR Character Set (E13B Font)

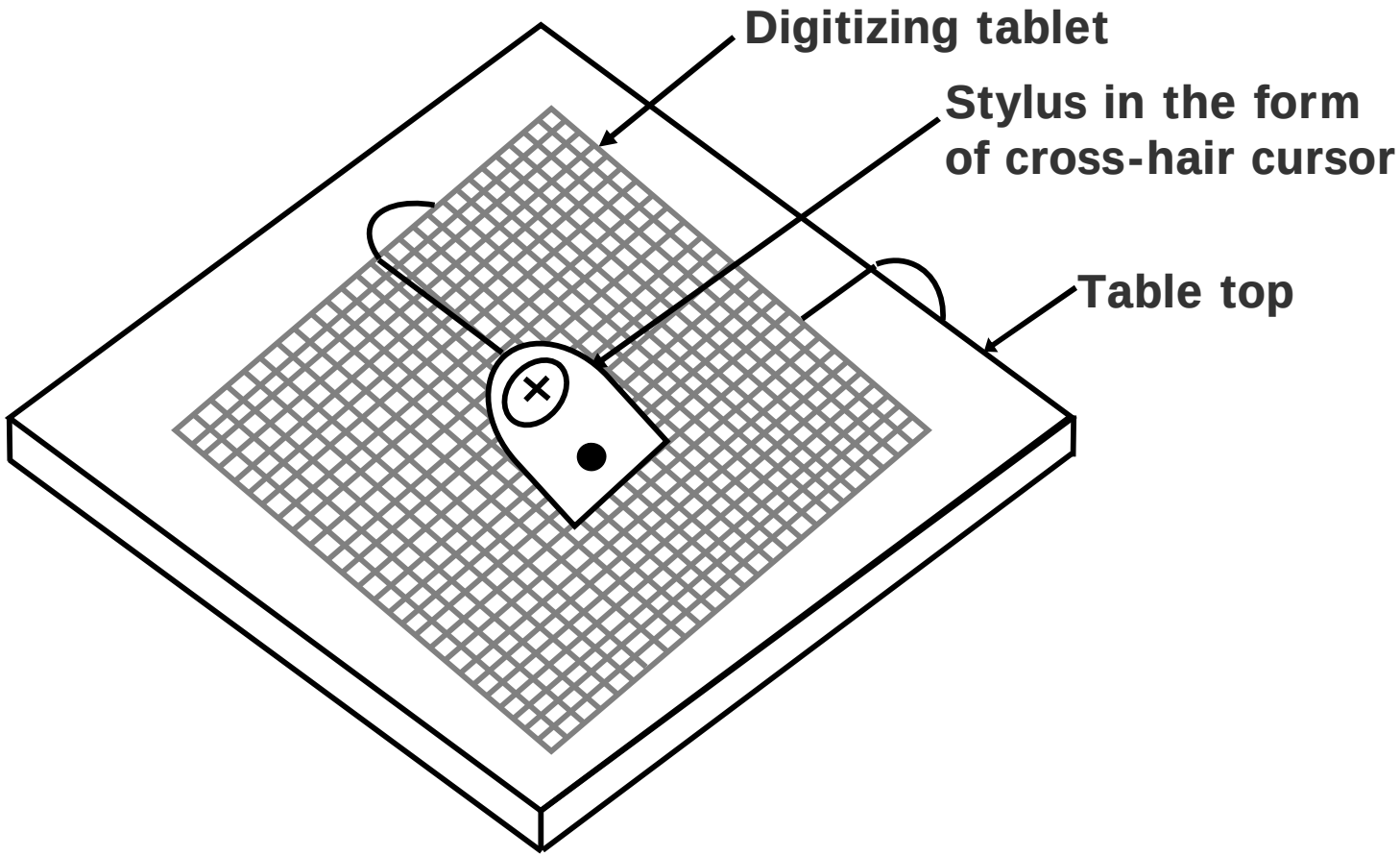


- It consists of numerals 0 to 9 and four special characters
- MICR is not adopted by other industries because it supports only 14 symbols

Digitizer

- Input device used for converting (digitizing) pictures, maps and drawings into digital form for storage in computers
- Commonly used in the area of Computer Aided Design (CAD) by architects and engineers to design cars, buildings medical devices, robots, mechanical parts, etc.
- Used in the area of Geographical Information System (GIS) for digitizing maps available in paper form

A Digitizer


Digitizing tablet
Stylus in the form of cross-hair cursor
Table top

Electronic-card Reader

- Electronic cards are small plastic cards having encoded data appropriate for the application for which they are used
- Electronic-card reader (normally connected to a computer) is used to read data encoded on an electronic card and transfer it to the computer for further processing
- Used together as a means of direct data entry into a computer system
- Used by banks for use in automatic teller machines (ATMs) and by organizations for controlling access of employees to physically secured areas

Speech Recognition Devices

- Input device that allows a person to input data to a computer system by speaking to it
- Today's speech recognition systems are limited to accepting few words within a relatively small domain and can be used to enter only limited kinds and quantities of data

Types of Speech Recognition Systems

(Continued from previous slide..)

- ***Single word recognition systems*** **can recognize only a single spoken words, such as YES, NO, MOVE, STOP, at a time. Speaker-independent systems are mostly of this type**
- ***Continuous speech recognition systems*** **can recognize spoken sentences, such as MOVE TO THE NEXT BLOCK. Such systems are normally speaker-dependent**

Uses of Speech Recognition Systems

- For inputting data to a computer system by a person in situations where his/her hands are busy, or his/her eyes must be fixed on a measuring instrument or some other object
- For data input by dictation of long text or passage for later editing and review
- For authentication of a user by a computer system based on voice input
- For limited use of computers by individuals with physical disabilities

Vision-Input Systems

- Allow computer to accept input just by seeing an object.
- Input data is normally an object's shape and features in the form of an image
- Mainly used today in factories for designing industrial robots that are used for quality-control and assembly processes

Commonly Used Output Devices

- Monitors
- Printers
- Plotters
- Screen image projector
- Voice response systems

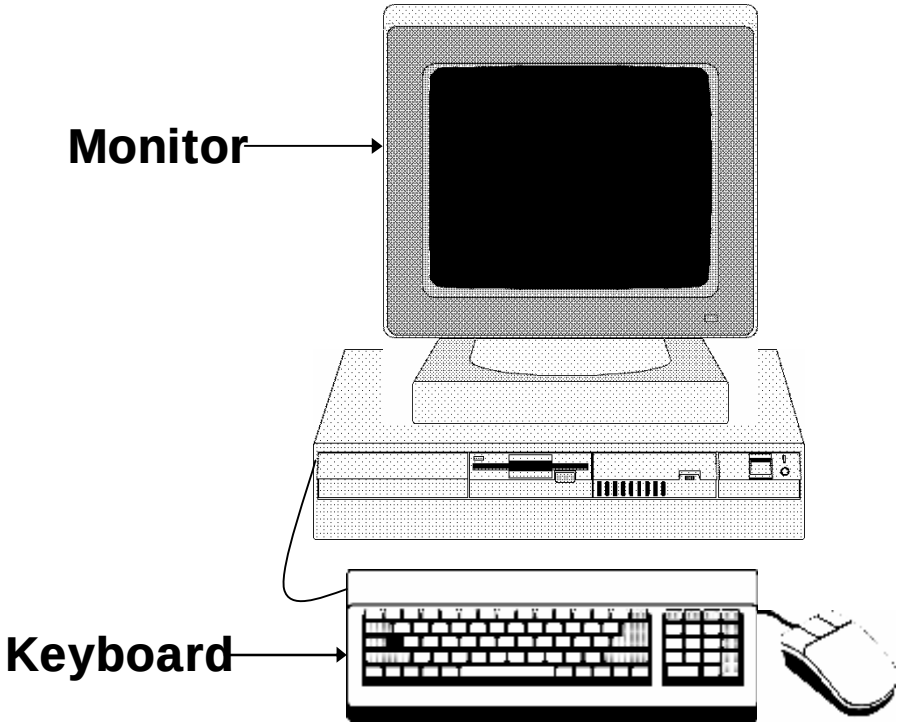
Types of Output

- Soft-copy output
  - Not produced on a paper or some material that can be touched and carried for being shown to others
  - Temporary in nature and vanish after use
  - Examples are output displayed on a terminal screen or spoken out by a voice response system
- Hard-copy output
  - Produced on a paper or some material that can be touched and carried for being shown to others
  - Permanent in nature and can be kept in paper files or can be looked at a later time when the person is not using the computer
  - Examples are output produced by printers or plotters on paper

Monitors

- Monitors are the most popular output devices used for producing soft-copy output
- Display the output on a television like screen
- Monitor associated with a keyboard is called a video display terminal (VDT). It is the most popular I/O device

Monitors



A video display terminal consists of a monitor and a keyboard

Types of Monitors

- Cathode-ray-tube (CRT) monitors look like a television and are normally used with non-portable computer systems
- Flat-panel monitors are thinner and lighter and are commonly used with portable computer systems like notebook computers. Now they are also used with non-portable desktop computer systems because they occupy less table space.

Printers

Most common output devices for producing hard-copy output

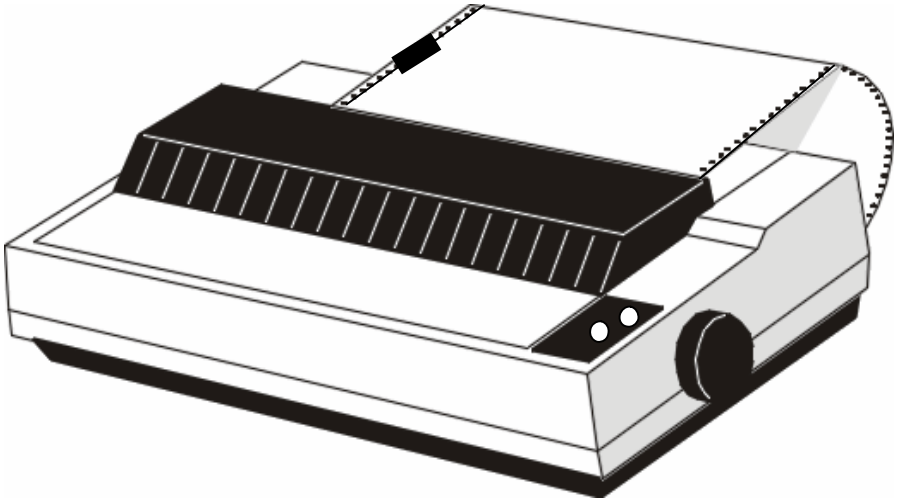
Dot-Matrix Printers

- Character printers that form characters and all kinds of images as a pattern of dots
- Print many special characters, different sizes of print and graphics such as charts and graphs
- Impact printers can be used for generating multiple copies by using carbon paper or its equivalent
- Slow, with speeds usually ranging between 30 to 600 characters per second
- Cheap in both initial cost and cost of operation

Formation of Characters as a pattern of dots

ABCDEFGHIJKLMNO
PQRSTUVWXYZ
0123456789-.,
&/$*#%@=(+)

A Dot Matrix Printer



Inkjet Printers

- Character printers that form characters and all kinds of images by spraying small drops of ink on to the paper
- Print head contains up to 64 tiny nozzles that can be selectively heated up in a few micro seconds by an integrated circuit register
- To print a character, the printer selectively heats the appropriate set of nozzles as the print head moves horizontally
- Can print many special characters, different sizes of print, and graphics such as charts and graphs

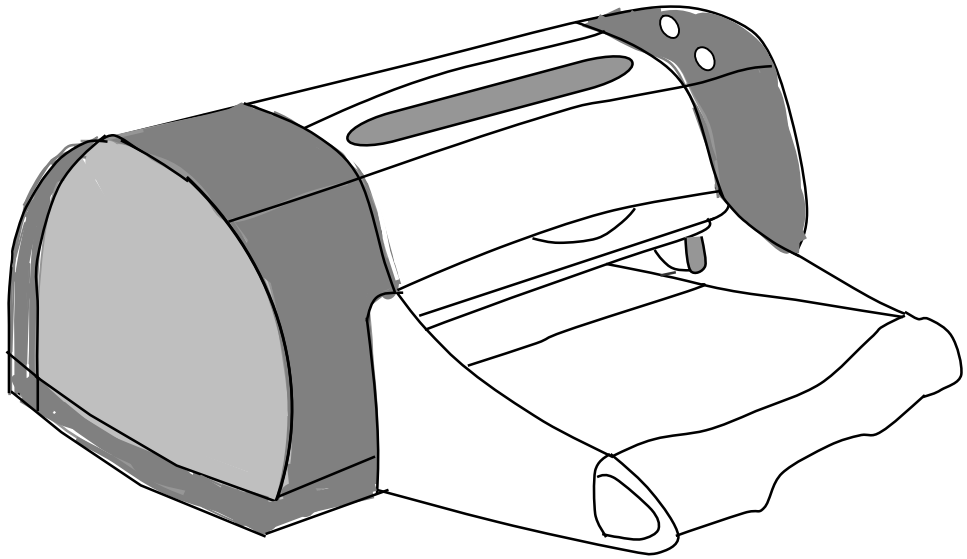
(Continued on next slide)

Inkjet Printers

(Continued from previous slide..)

- Non-impact printers. Hence, they cannot produce multiple copies of a document in a single printing
- Can be both monochrome and color
- Slower than dot-matrix printers with speeds usually ranging between 40 to 300 characters per second
- More expensive than a dot-matrix printer

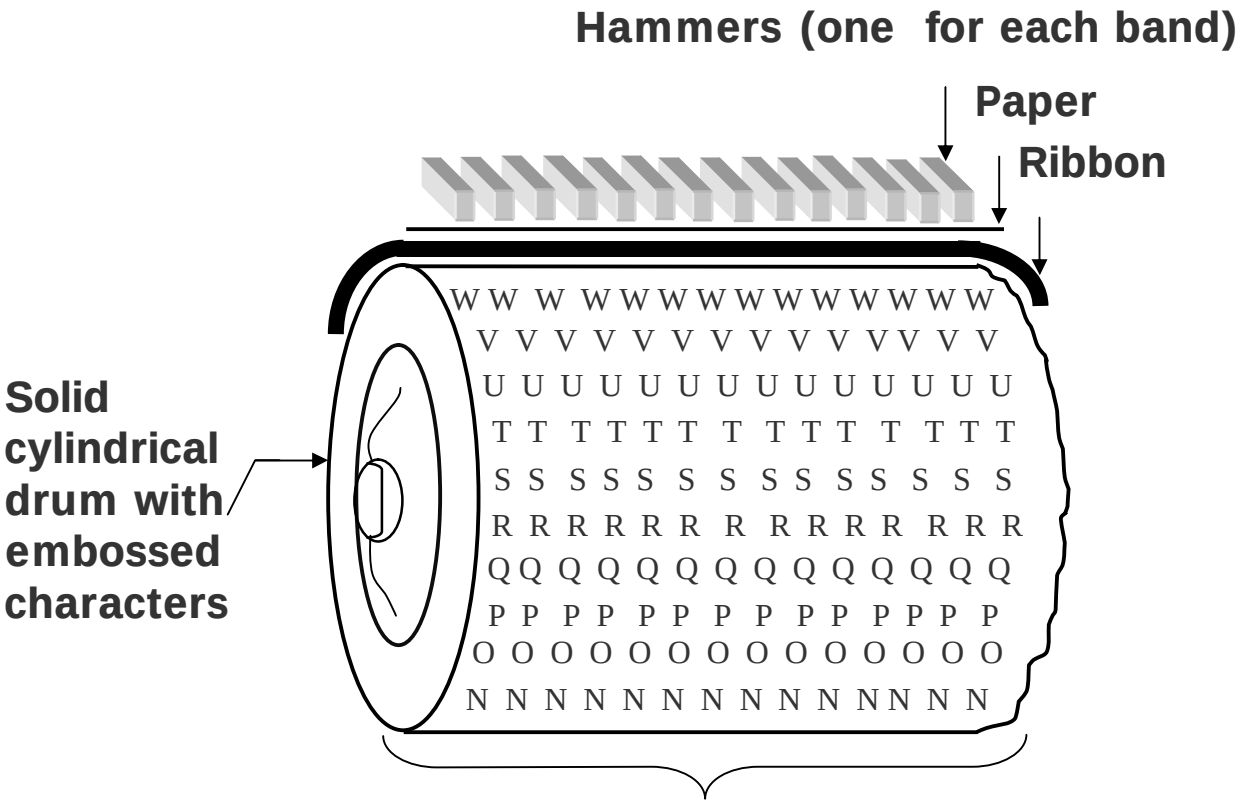
An Inkjet Printers



Drum Printers

- Line printers that print one line at a time
- Have a solid cylindrical drum with characters embossed on its surface in the form of circular bands
- Set of hammers mounted in front of the drum in such a manner that an inked ribbon and paper can be placed between the hammers and the drum
- Can only print a pre-defined set of characters in a pre-defined style that is embossed on the drum
- Impact printers and usually monochrome
- Typical speeds are in the range of 300 to 2000 lines per minute

Printing Mechanism of a Drum Printer



Chain / Band Printers

- Line printers that print one line at a time
- Consist of a metallic chain/band on which all characters of the character set supported by the printer are embossed
- Also have a set of hammers mounted in front of the chain/band in such a manner that an inked ribbon and paper can be placed between the hammers and the chain/band

Chain / Band Printers

- Can only print pre-defined sets of characters that are embossed on the chain/band used with the printer
- Cannot print any shape of characters, different sizes of print, and graphics such as charts and graphs
- Are impact printers and can be used for generating multiple copies by using carbon paper or its equivalent
- Are usually monochrome
- Typical speeds are in the range of 400 to 3000 lines per minute

Laser Printers

- Page printers that print one page at a time
- Consist of a laser beam source, a multi-sided mirror, a photoconductive drum and toner (tiny particles of oppositely charged ink)
- To print a page, the laser beam is focused on the electro statically charged drum by the spinning multi-sided mirror
- Toner sticks to the drum in the places the laser beam has charged the drum's surface.
- Toner is then permanently fused on the paper with heat and pressure to generate the printer output
- Laser printers produce very high quality output having resolutions in the range of 600 to 1200 dpi

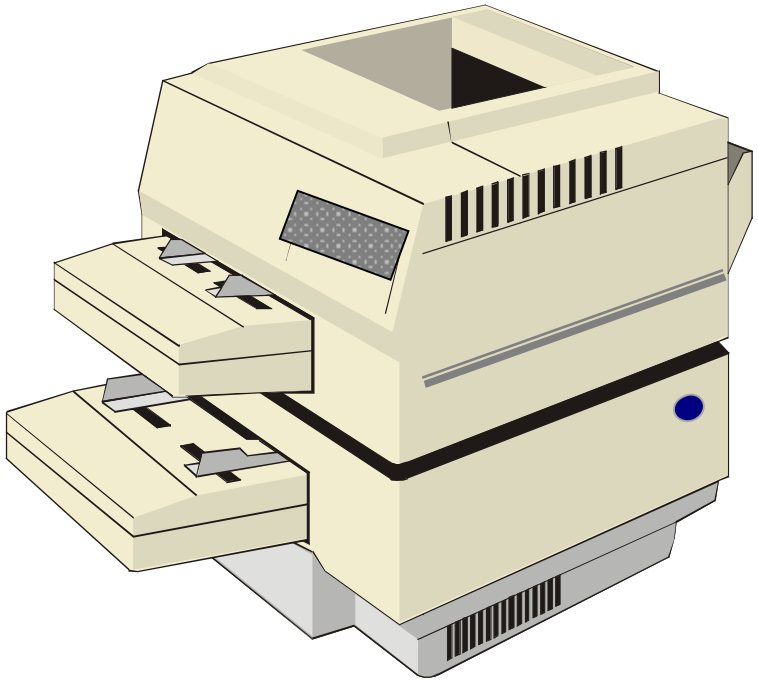
(Continued on next slide)

Laser Printers

(Continued from previous slide..)

- Can print many special characters, different sizes of print, and graphics such as charts and graphs
- Are non-impact printers
- Most laser printers are monochrome, but color laser printers are also available
- Low speed laser printers can print 4 to 12 pages per minute. Very high-speed laser printers can print 500 to 1000 pages per minute
- More expensive than other printers

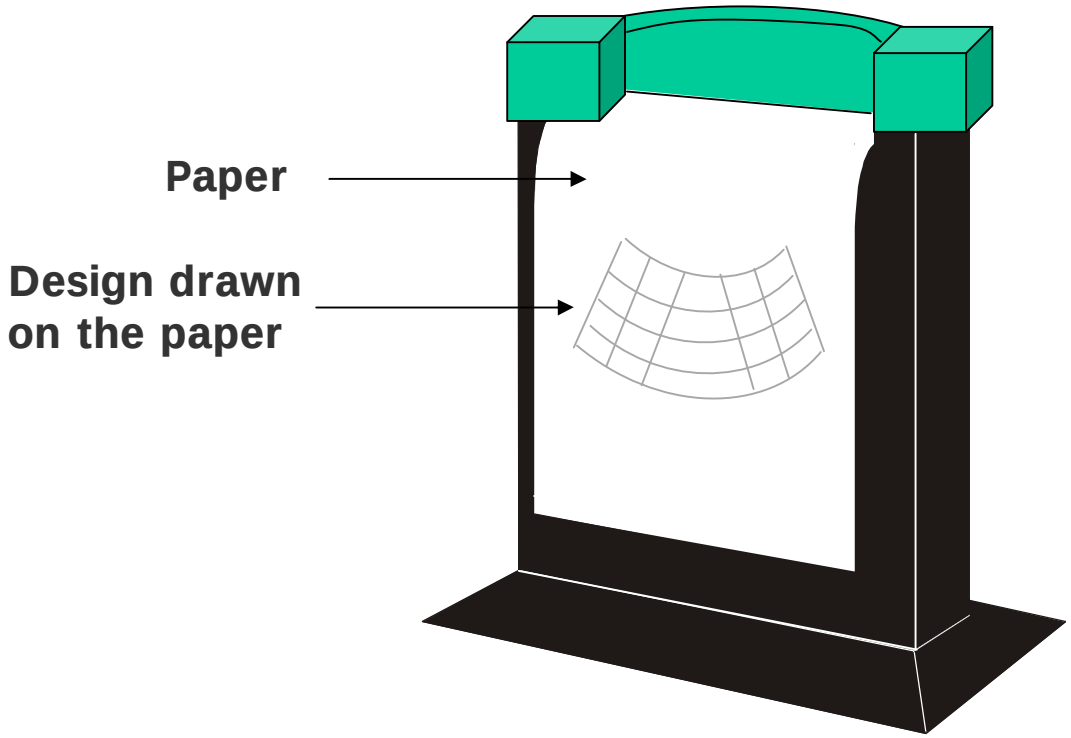
A Laser Printers



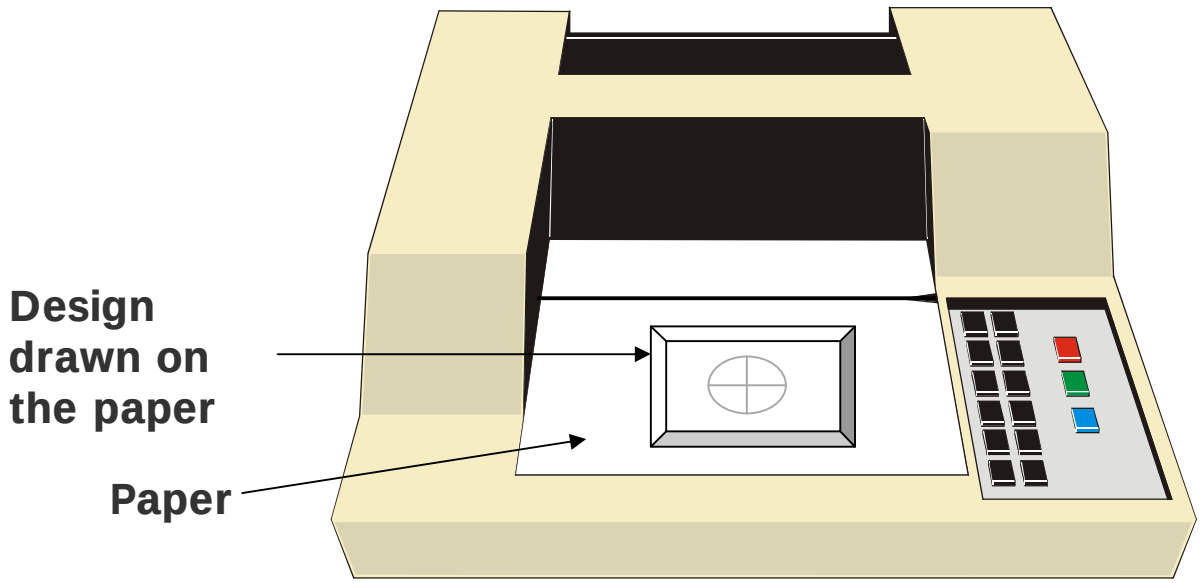
Plotters

- Plotters are an ideal output device for architects, engineers, city planners, and others who need to routinely generate high-precision, hard-copy graphic output of widely varying sizes
- Two commonly used types of plotters are:
  - *Drum plotter*, in which the paper on which the design has to be made is placed over a drum that can rotate in both clockwise and anti-clockwise directions
  - *Flatbed plotter*, in which the paper on which the design has to be made is spread and fixed over a rectangular flatbed table

A Drum Plotter



A Flatbed Plotter



Screen Image Projector

- An output device that can be directly plugged to a computer system for projecting information from a computer on to a large screen
- Useful for making presentations to a group of people with direct use of a computer
- Full-fledged multimedia presentation with audio, video, image, and animation can be prepared and made using this facility

Voice Response Systems

- Voice response system enables a computer to talk to a user
- Has an audio-response device that produces audio output
- Such systems are of two types:
  - Voice reproduction systems
  - Speech synthesizers

(Continued on next slide)

Voice Reproduction Systems

(Continued from previous slide..)

- Produce audio output by selecting an appropriate audio output from a set of pre-recorded audio responses
- Applications include audio help for guiding how to operate a system, automatic answering machines, video games, etc.

Speech Synthesizers

- Converts text information into spoken sentences
- Used for applications such as:
  - Reading out text information to blind persons
  - Allowing those persons who cannot speak to communicate effectively
  - Translating an entered text into spoken words in a selected language

Key Words/Phrases

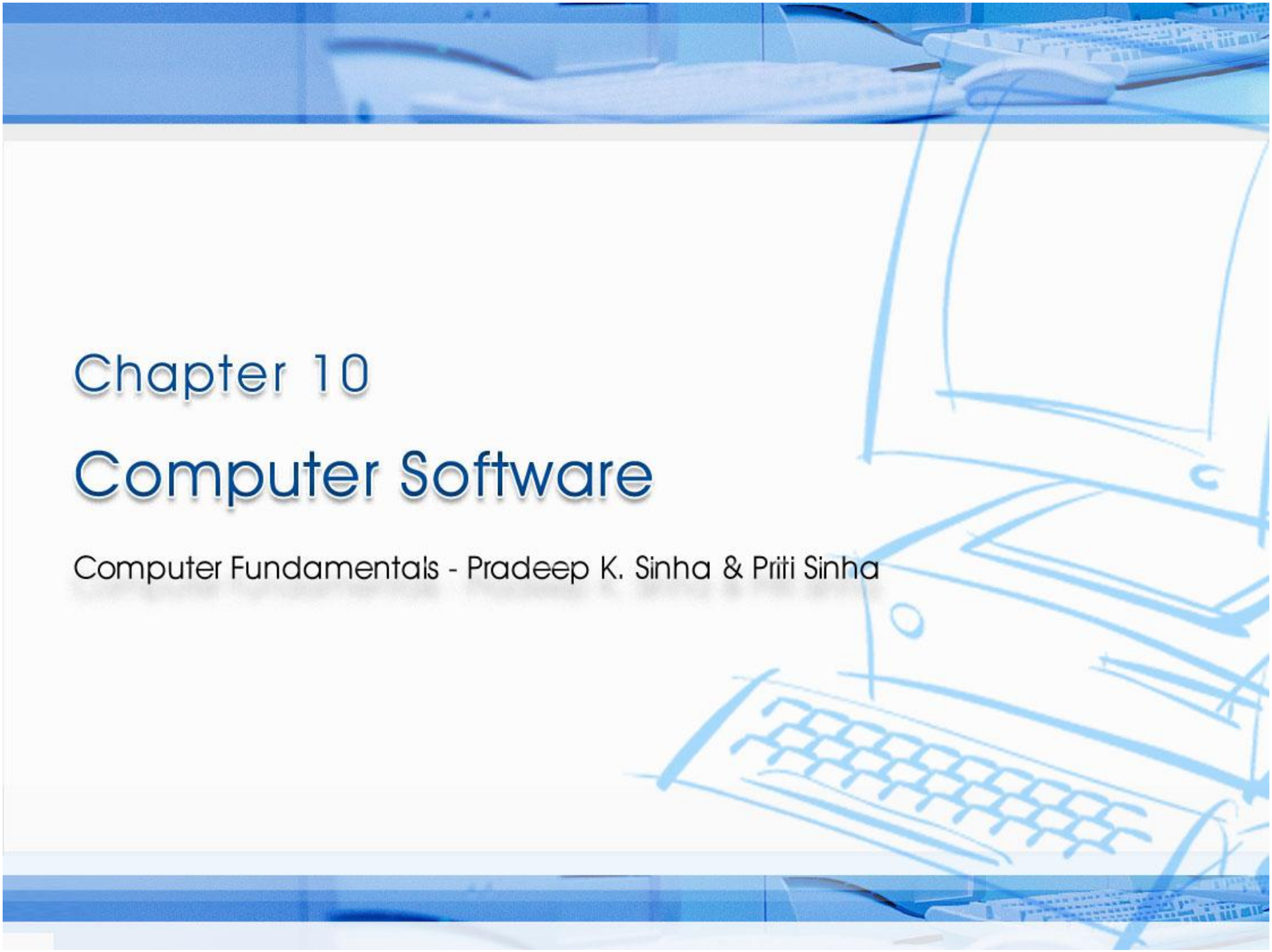
- Bard code reader
- Cathode Ray Tube (CRT)
- Chain/Band printer
- Data scanning device
- Digitizer
- Digitizing tablet
- Dot-Matrix printer
- Drum plotter
- Drum printer
- Electronic card reader
- Electronic Pen
- Flatbed plotter
- Flatbed Scanner
- Graphical User Interface
- Hand-held scanner
- Hard-copy output
- Image Scanner
- Information Kiosk
- Inkjet printer
- Input/Output device
- Joystick
- Keyboard device
- Laser printer
- Magnetic-Ink Character Recognition (MICR)
- Monitor
- Mouse
- Optical Character Recognition (OCR)
- Optical Mark Reader (OMR)
- Peripheral device
- Phonemes
- Plotter
- Point-and-draw device
- Printer
- QWERTY keyboard
- Screen Image Projector

(Continued on next slide)

Key Words / Phrases

(Continued from previous slide..)

- Soft-copy output
- Speech synthesizer
- Stylus
- Touch Screen
- Trackball
- Universal Product Code (UPC)
- Video Display Terminal (VDT)
- Vision-input system
- Voice recognition device
- Voice reproduction system
- Voice response system



Chapter 10

Computer Software

Computer Fundamentals - Pradeep K. Sinha & Priti Sinha

Learning Objectives

In this chapter you will learn about:

- Term "Software" and its relationship with "Hardware"
- Various types of software and their examples
- Relationship among hardware, system software, application software, and users of a computer system
- Different ways of acquiring software
- Various steps involved in software development
- Firmware
- Middleware

Software

- ***Hardware*** **refers to the physical devices of a computer system.**
- ***Software*** **refers to a collection of programs**
- ***Program*** **is a sequence of instructions written in a language that can be understood by a computer**
- ***Software package*** **is a group of programs that solve a specific problem or perform a specific type of job**

Relationship Between Hardware and Software

- Both hardware and software are necessary for a computer to do useful job. They are complementary to each other
- Same hardware can be loaded with different software to make a computer system perform different types of jobs
- Except for *upgrades*, hardware is normally a one-time expense, whereas software is a continuing expense
- Upgrades refer to renewing or changing components like increasing the main memory, or hard disk capacities, or adding speakers, modems, etc.

Types of Software

Most software can be divided into two major categories:

- *System software* are designed to control the operation and extend the processing capability of a computer system
- *Application software* are designed to solve a specific problem or to do a specific task

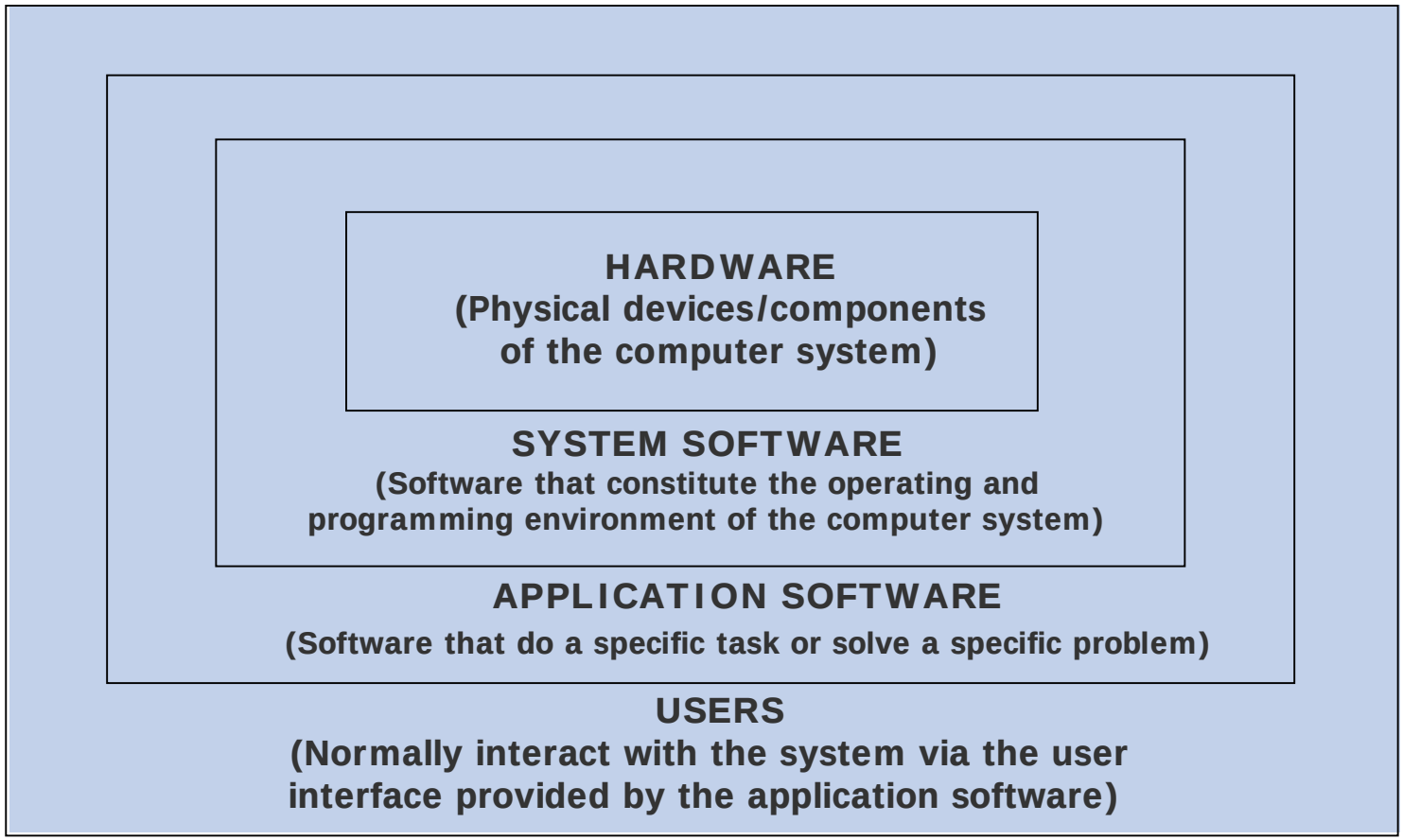
System Software

- Make the operation of a computer system more effective and efficient
- Help hardware components work together and provide support for the development and execution of application software
- Programs included in a system software package are called *system programs* and programmers who prepare them are called *system programmers*
- Examples of system software are operating systems, programming language translators, utility programs, and communications software

Application Software

- Solve a specific problem or do a specific task
- Programs included in an application software package are called *application programs* and the programmers who prepare them are called *application programmers*
- Examples of application software are word processing, inventory management, preparation of tax returns, banking, etc.

Logical System Architecture



Relationship among hardware, system software, application software, and users of a computer system.

Ways of Acquiring Software

- Buying pre-written software
- Ordering customized software
- Developing customized software
- Downloading public-domain software

Each of these ways of acquiring software has its own advantages and limitations

Advantages and Limitations of Buying Pre-written Software

- Usually costs less
- Planned activity can be stared almost immediately
- Often, operating efficiency and the capability to meet specific needs of user more effectively in not as good for pre-written software packages as for in-house developed software packages

Advantages & Limitations of Ordering Customized Software

- User need not maintain its own software development team, which is an expensive affair
- User needs to always depend on the vendor for carrying out the changes and the vendor may separately charge for every request for change

Advantages & Limitations of Developing Customized Software

- Easier to carry out changes in the software, if it is developed in-house
- Developing software in-house means a major commitment of time, money, and resources
- In-house software development team needs to be maintained and managed

Advantage & Limitations of Downloading Public-domain Software

- Available for free or as shareware, and are usually accompanied with source code
- Usually community-supported as author does not support users directly
- Can be downloaded and used immediately
- They may not be properly tested before release
- Open Source Software (OSS) are becoming popular due to:
  - Allows any user to download, view, modify, and redistribute
  - User can fix bugs or change software to suit needs
  - Copyright is protected for both original and subsequent authors
- Not all open source software are free and vise-verse

Software Development Steps

Developing a software and putting it to use is a complex process and involves following steps:

- Analyzing the problem at hand and planning the program(s) to solve the problem
- Coding the program(s)
- Testing, debugging, and documenting the program(s)
- Implementing the program(s)
- Evaluating and maintaining the program(s)

Firmware

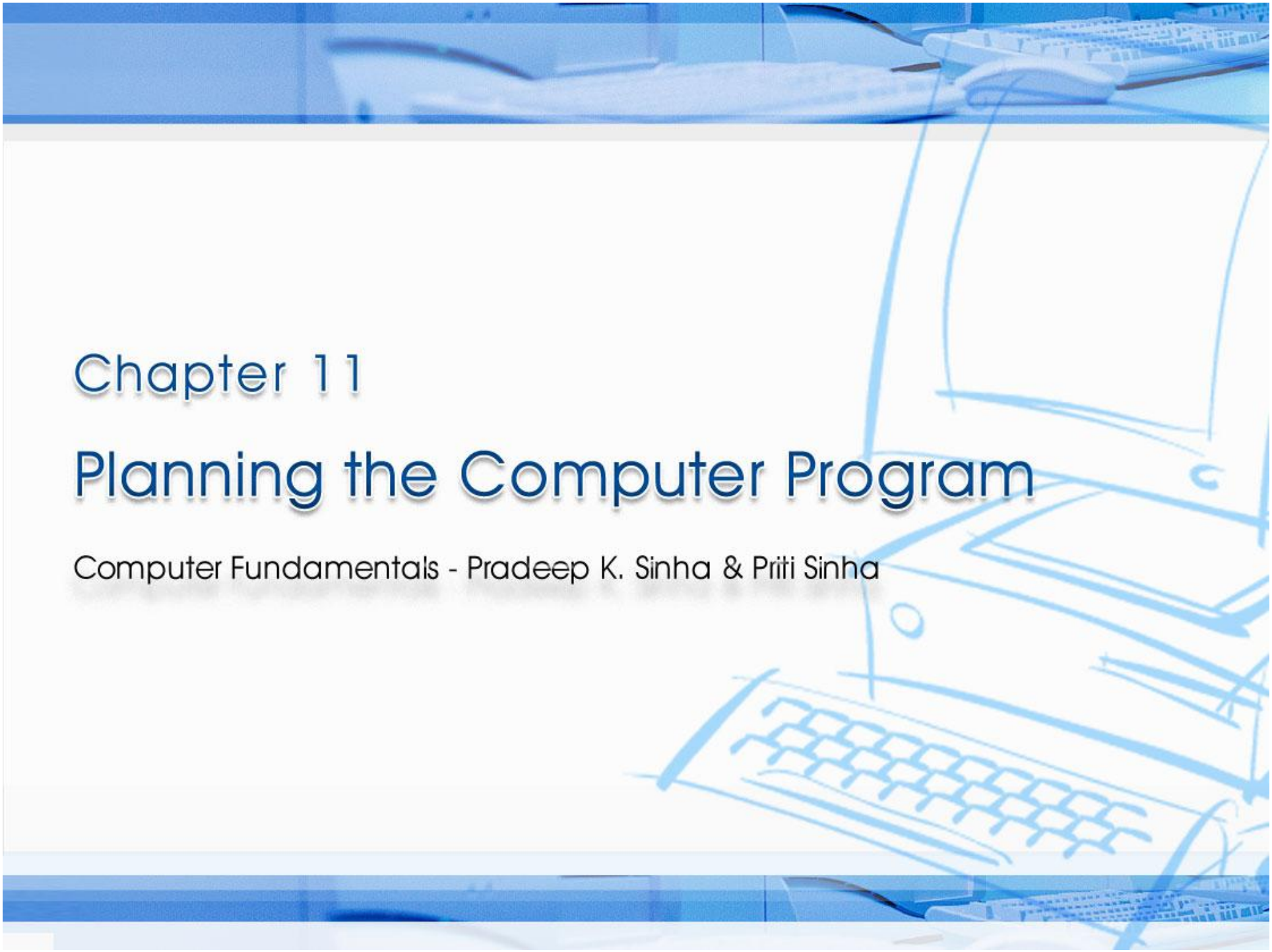
- Firmware is software substituted for hardware and stored in read-only memory
- Firmware technology has enabled production of various types of smart machines having microprocessor chips with embedded software

Middleware

- Basic idea is to have a *separate software layer* to:
  - Act as “glue” between client and server parts of application
  - Provide programming abstraction
  - Mask heterogeneity of underlying network, hardware, and OS
- Encourages *three-tier* software architecture against two-tier popularized by Server-Client architecture

Key Words / Phrases

- Application programmers
- Application programs
- Application software
- Computer program
- Customized software
- Database
- Education software
- End-to-end solution
- Entertainment software
- Firmware
- Graphics software
- Hardware
- Middleware
- Open Source Software
- Personal assistance software
- Pre-written software
- Public-domain software
- Shareware
- Software
- Software package
- Spreadsheet
- System programmers
- System programs
- System software
- Turnkey solution
- User-supported software
- Utilities
- Word-processing



Chapter 11

Planning the Computer Program

Computer Fundamentals - Pradeep K. Sinha & Priti Sinha

Learning Objectives

In this chapter you will learn about:

- Programs must be planned before they are written
- Algorithm
- Flowchart
- Pseudocode
- Plan the logic of a computer program
- Commonly used tools for program planning and their use

Purpose of Program Planning

- To write a correct program, a programmer must write each and every instruction in the correct sequence
- Logic (instruction sequence) of a program can be very complex
- Hence, programs must be planned before they are written to ensure program instructions are:
  - Appropriate for the problem
  - In the correct sequence

Algorithm

- Refers to the logic of a program and a step-by-step description of how to arrive at the solution of a given problem
- In order to qualify as an algorithm, a sequence of instructions must have following characteristics:
  - Each and every instruction should be precise and unambiguous
  - Each instruction should be such that it can be performed in a finite time
  - One or more instructions should not be repeated infinitely. This ensures that the algorithm will ultimately terminate
  - After performing the instructions, that is after the algorithm terminates, the desired results must be obtained

Sample Algorithm (Example 1)

There are 50 students in a class who appeared in their final examination. Their mark sheets have been given to you.

The division column of the mark sheet contains the division (FIRST, SECOND, THIRD or FAIL) obtained by the student.

Write an algorithm to calculate and print the total number of students who passed in FIRST division.

Sample Algorithm (Example 1)

(contd...)

Step 1: Initialize Total_First_Division and Total_Marksheets_Checked to zero.

Step 2: Take the mark sheet of the next student.

Step 3: Check the division column of the mark sheet to see if it is FIRST, if no, go to Step 5.

Step 4: Add 1 to Total_First_Division.

Step 5: Add 1 to Total_Marksheets_Checked.

Step 6: Is Total_Marksheets_Checked = 50, if no, go to Step 2.

Step 7: Print Total_First_Division.

Step 8: Stop.

Sample Algorithm (Example 2)

There are 100 employees in an organization. The organization wants to distribute annual bonus to the employees based on their performance. The performance of the employees is recorded in their annual appraisal forms.

Every employee's appraisal form contains his/her basic salary and the grade for his/her performance during the year. The grade is of three categories – 'A' for outstanding performance, 'B' for good performance, and 'C' for average performance.

It has been decided that the bonus of an employee will be 100% of the basic salary for outstanding performance, 70% of the basic salary for good performance, 40% of the basic salary for average performance, and zero for all other cases.

Write an algorithm to calculate and print the total bonus amount to be distributed by the organization.

Sample Algorithm (Example 2)

(contd...)

Step 1: Initialize Total_Bonus and Total_Employees_Checked to zero.

Step 2: Initialize Bonus and Basic_Salary to zero.

Step 3: Take the appraisal form of the next employee.

Step 4: Read the employee's Basic_Salary and Grade.

Step 5: If Grade = A, then Bonus = Basic_Salary. Go to Step 8.

Step 6: If Grade = B, then Bonus = Basic_Salary x 0.7. Go to Step 8.

Step 7: If Grade = C, then Bonus = Basic_Salary x 0.4.

Step 8: Add Bonus to Total_Bonus.

Step 9: Add 1 to Total_Employees_Checked.

Step 10: If Total_Employees_Checked < 100, then go to Step 2.

Step 11: Print Total_Bonus.

Step 12: Stop.

Representation of Algorithms

- As programs
- As flowcharts
- As pseudocodes

When an algorithm is represented in the form of a programming language, it becomes a program

Thus, any program is an algorithm, although the reverse is not true

Flowchart

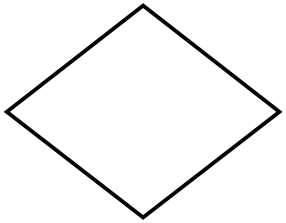
- *Flowchart* is a pictorial representation of an algorithm
- Uses symbols (boxes of different shapes) that have standardized meanings to denote different types of instructions
- Actual instructions are written within the boxes
- Boxes are connected by solid lines having arrow marks to indicate the exact sequence in which the instructions are to be executed
- Process of drawing a flowchart for an algorithm is called *flowcharting*

Basic Flowchart Symbols

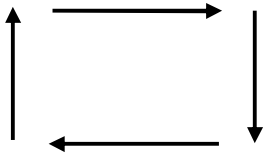
Terminal

Input/Output

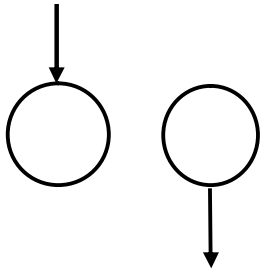
Processing



Decision

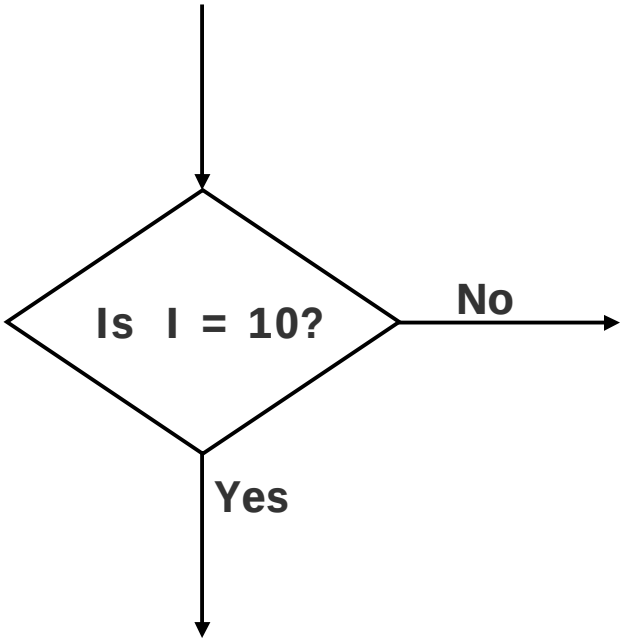


Flow lines

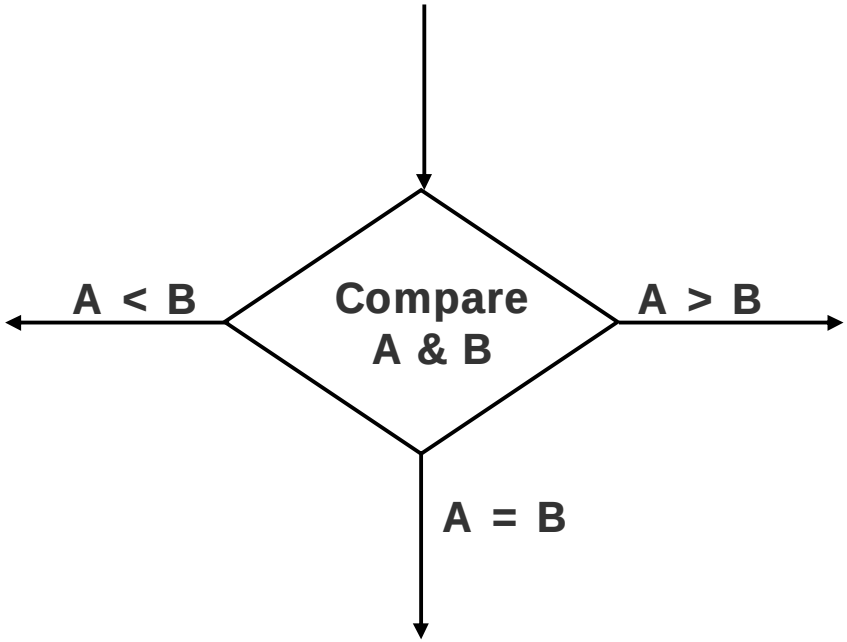


Connectors

Examples of Decision Symbol



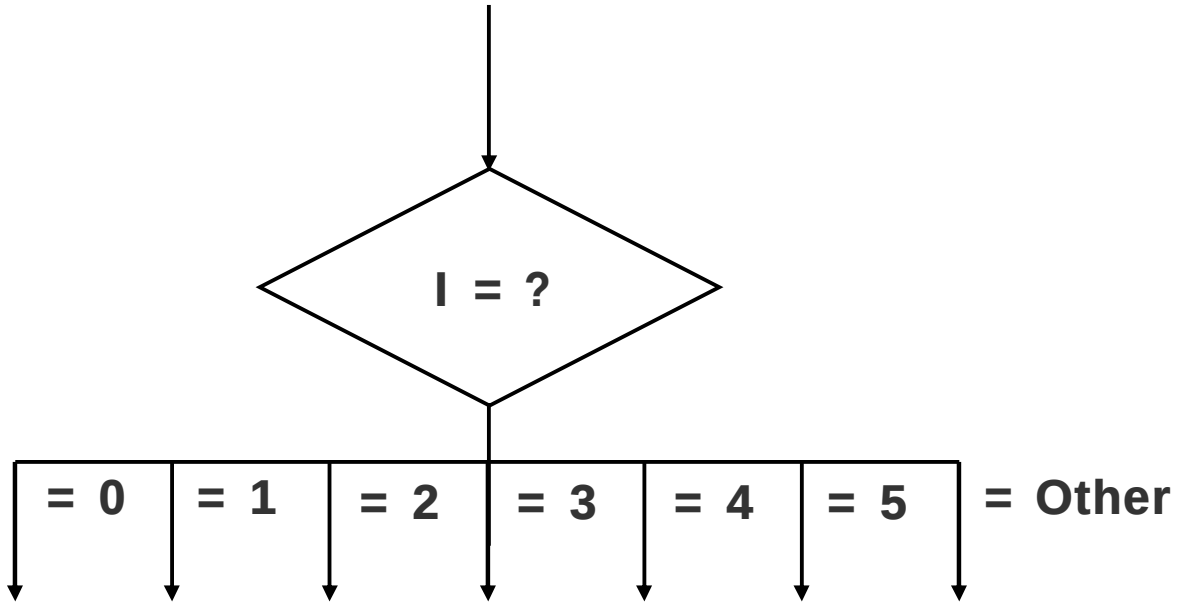
(a) A two-way branch decision.



(b) A three-way branch decision.

Examples of Decision Symbol

(contd...)



(c) A multiple-way branch decision.

Sample Flowchart (Example 3)

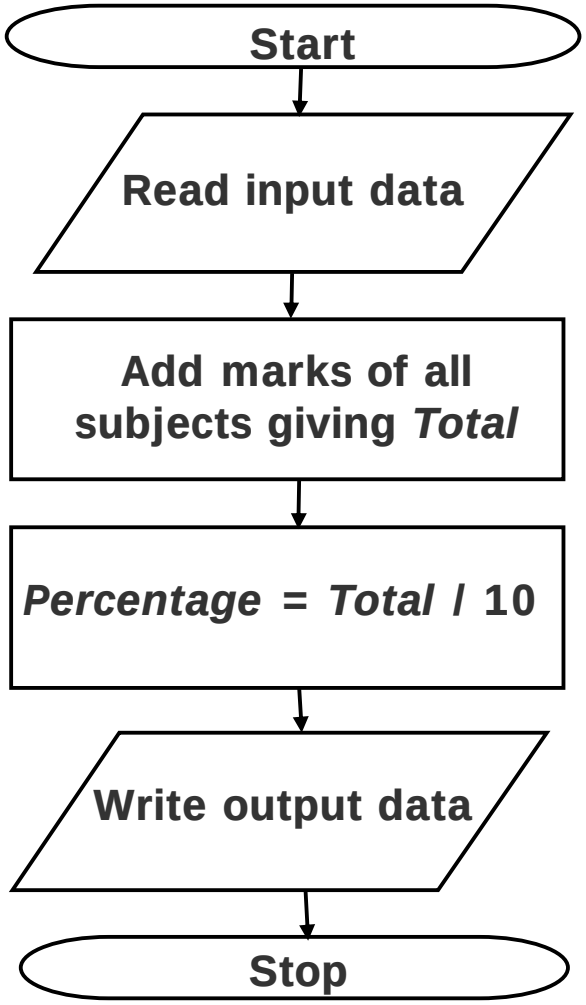
A student appears in an examination, which consists of total 10 subjects, each subject having maximum marks of 100.

The roll number of the student, his/her name, and the marks obtained by him/her in various subjects are supplied as input data.

Such a collection of related data items, which is treated as a unit is known as a record.

Draw a flowchart for the algorithm to calculate the percentage marks obtained by the student in this examination and then to print it along with his/her roll number and name.

Sample Flowchart (Example 3)



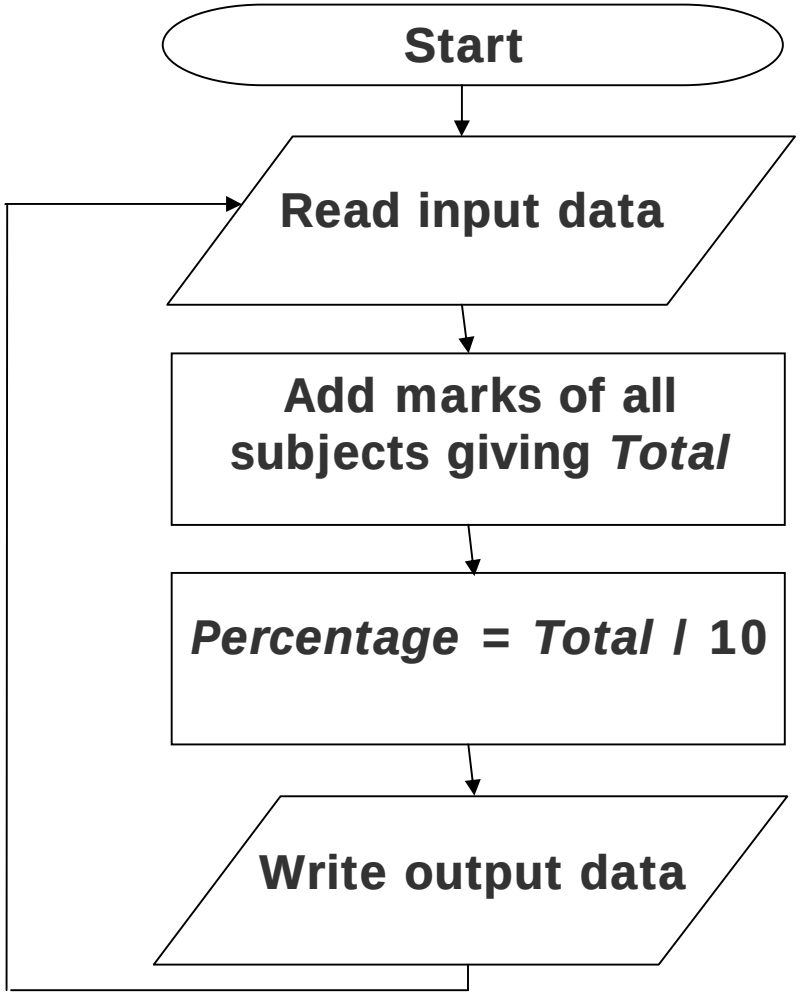
(contd...)

Sample Flowchart (Example 4)

50 students of a class appear in the examination of Example 3.

Draw a flowchart for the algorithm to calculate and print the percentage marks obtained by each student along with his/her roll number and name.

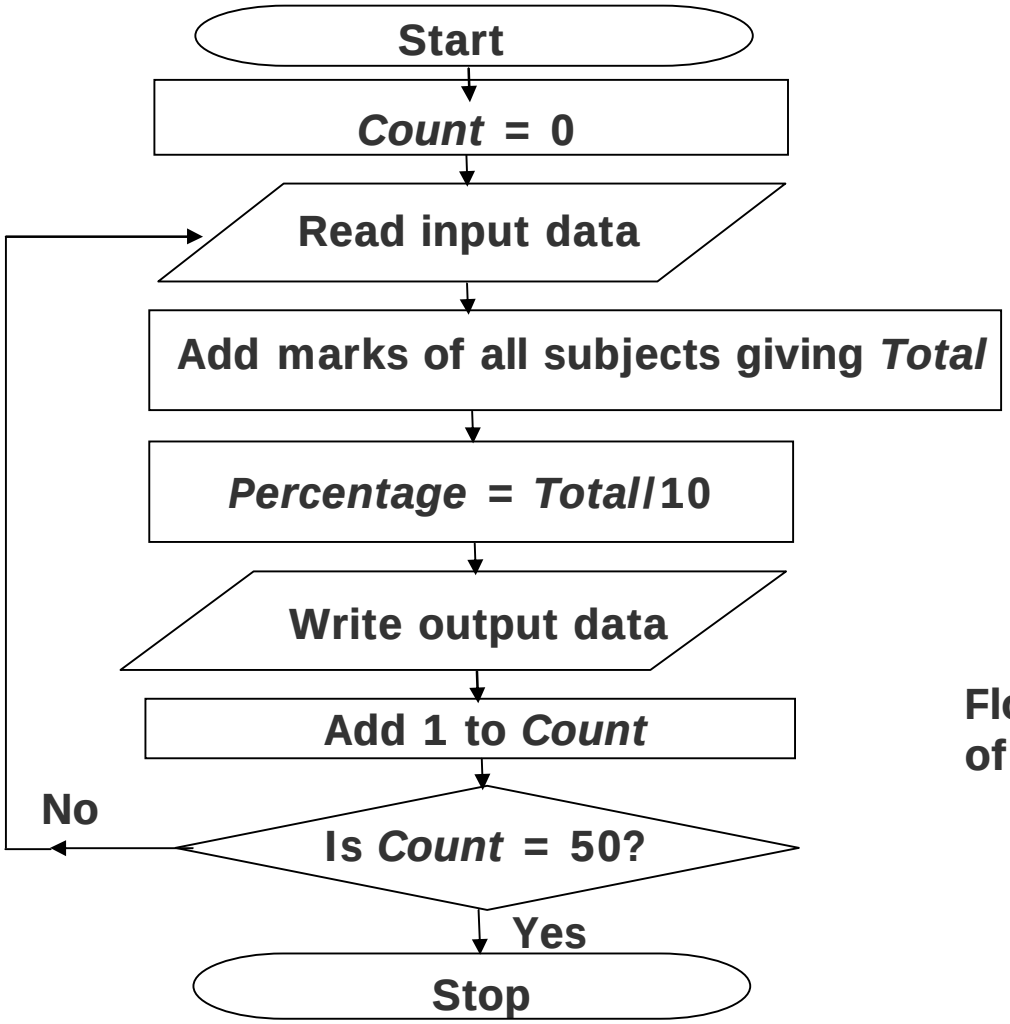
Sample Flowchart (Example 4)



(contd...)

Flowchart for the solution of Example 4 with an infinite (endless) process loop.

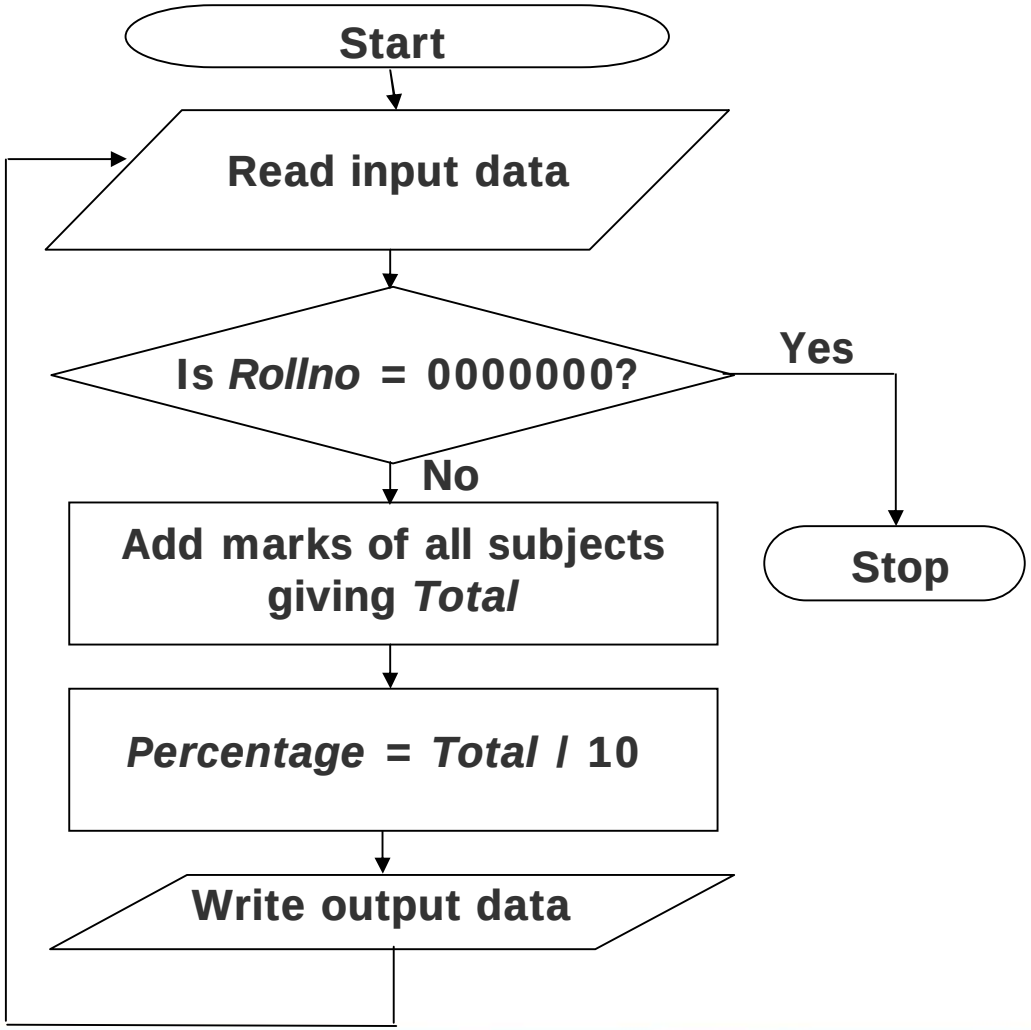
Sample Flowchart (Example 4)



(contd...)

Flowchart for the solution of Example 4.

Sample Flowchart (Example 4)



(contd...)

Generalized flowchart for the solution of Example 4 using the concept of trailer record. Here the process loop is terminated by detecting a special non-data record.

Sample Flowchart (Example 5)

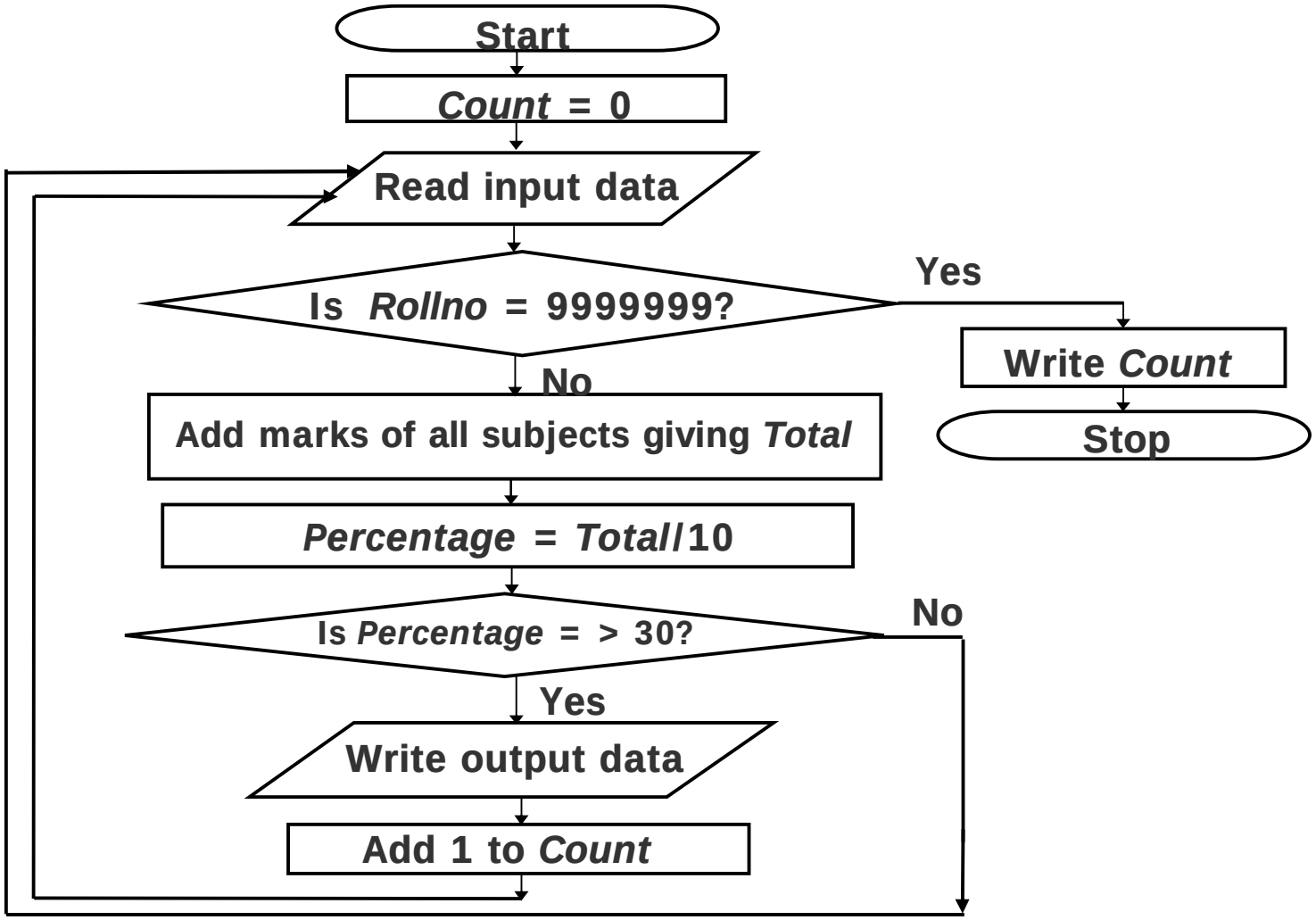
For the examination of Example 3, we want to make a list of only those students who have passed (obtained 30% or more marks) in the examination.

In the end, we also want to print out the total number of students who have passed.

Assuming that the input data of all the students is terminated by a trailer record, which has sentinel value of 9999999 for Rollno, draw a flowchart for the algorithm to do this.

Sample Flowchart (Example 5)

(contd...)



Sample Flowchart (Example 6)

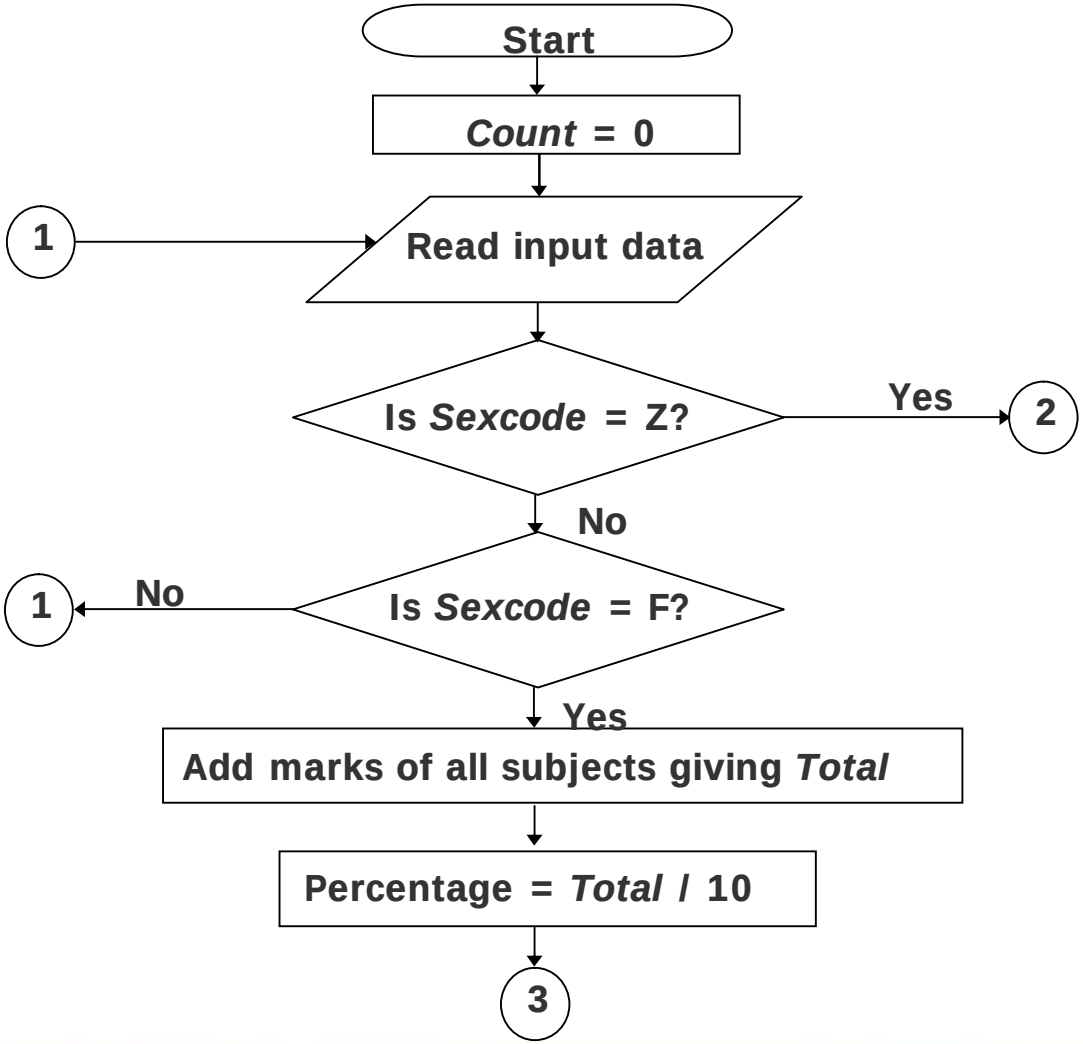
Suppose the input data of each student for the examination of Example 3 also contains information regarding the sex of the candidate in the field named *Sexcode* having values M (for male) or F (for female).

We want to make a list of only those female students who have passed in second division (obtained 45% or more but less than 60% marks).

In the end, we also want to print out the total number of such students.

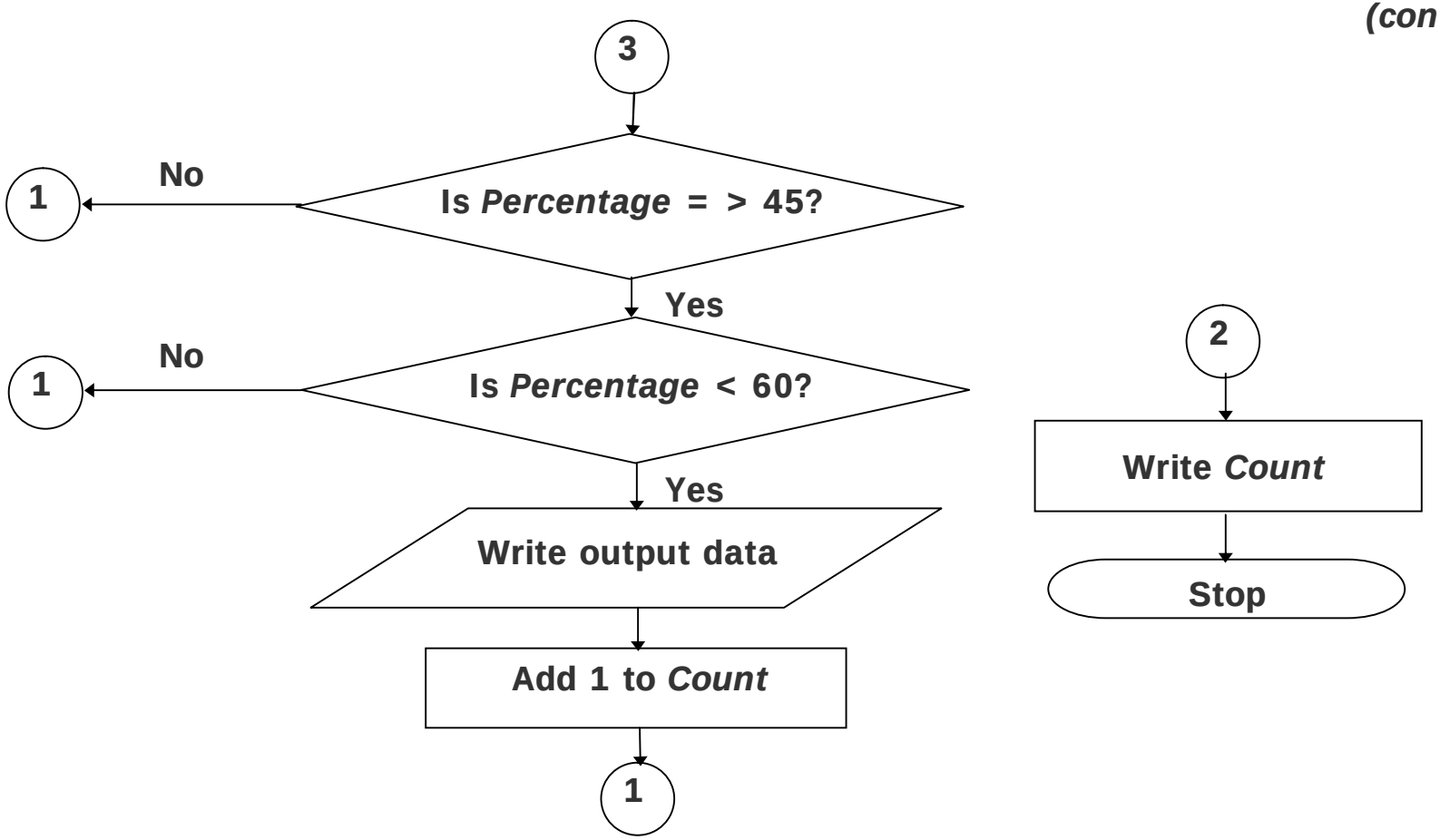
Assuming that the input data of all the students is terminated by a trailer record, which has a sentinel value of Z for Sexcode, draw a flowchart for the algorithm to do this.

Sample Flowchart (Example 6)



Sample Flowchart (Example 4)

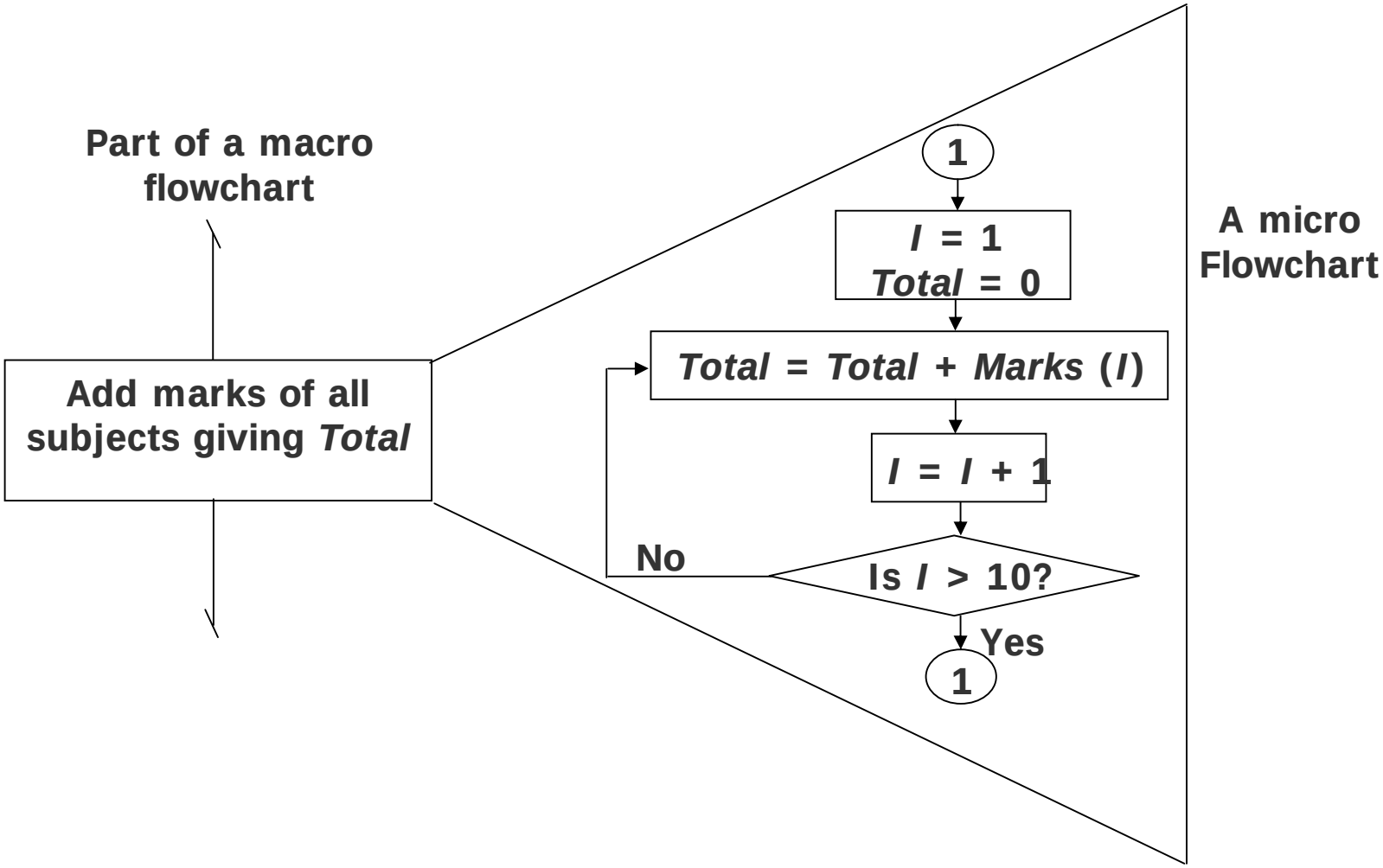
(contd...)



Levels of Flowchart

- Flowchart that outlines the main segments of a program or that shows less details is a *macro flowchart*
- Flowchart with more details is a *micro flowchart*, or detailed flowchart
- There are no set standards on the amount of details that should be provided in a flowchart

Example of Micro Flowchart



Flowcharting Rules

- First chart the main line of logic, then incorporate detail
- Maintain a consistent level of detail for a given flowchart
- Do not chart every detail of the program. A reader who is interested in greater details can refer to the program itself
- Words in the flowchart symbols should be common statements and easy to understand

Flowcharting Rules

(contd...)

- Be consistent in using names and variables in the flowchart
- Go from left to right and top to bottom in constructing flowcharts
- Keep the flowchart as simple as possible. Crossing of flow lines should be avoided as far as practicable
- If a new flowcharting page is needed, it is recommended that the flowchart be broken at an input or output point.
- Properly labeled connectors should be used to link the portions of the flowchart on different pages

Advantages of Flowchart

- Better Communication
- Proper program documentation
- Efficient coding
- Systematic debugging
- Systematic testing

Limitations of Flowchart

- Flowcharts are very time consuming and laborious to draw (especially for large complex programs)
- Redrawing a flowchart for incorporating changes/ modifications is a tedious task
- There are no standards determining the amount of detail that should be included in a flowchart

Pseudocode

- A program planning tool where program logic is written in an ordinary natural language using a structure that resembles computer instructions
- " Pseudo" means imitation or false and " Code" refers to the instructions written in a programming language. Hence, pseudocode is an imitation of actual computer instructions
- Because it emphasizes the design of the program, pseudocode is also called *Program Design Language (PDL)*

Basic Logic (Control) Structures

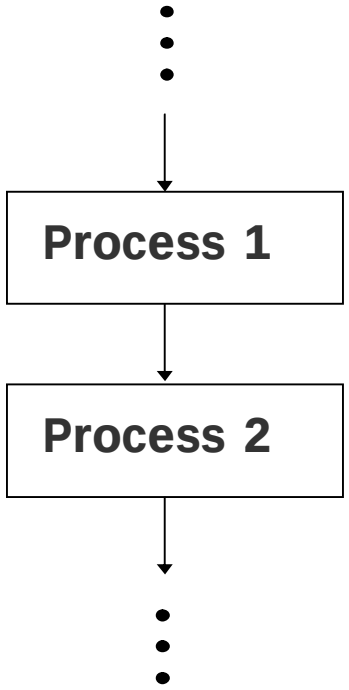
Any program logic can be expressed by using only following three simple logic structures:

1. Sequence logic,
2. Selection logic, and
3. Iteration (or looping) logic

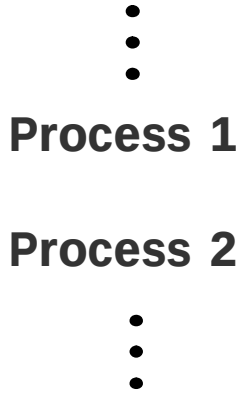
Programs structured by using only these three logic structures are called ***structured programs***, and the technique of writing such programs is known as ***structured programming***

Sequence Logic

It is used for performing instructions one after another in sequence.



(a) Flowchart



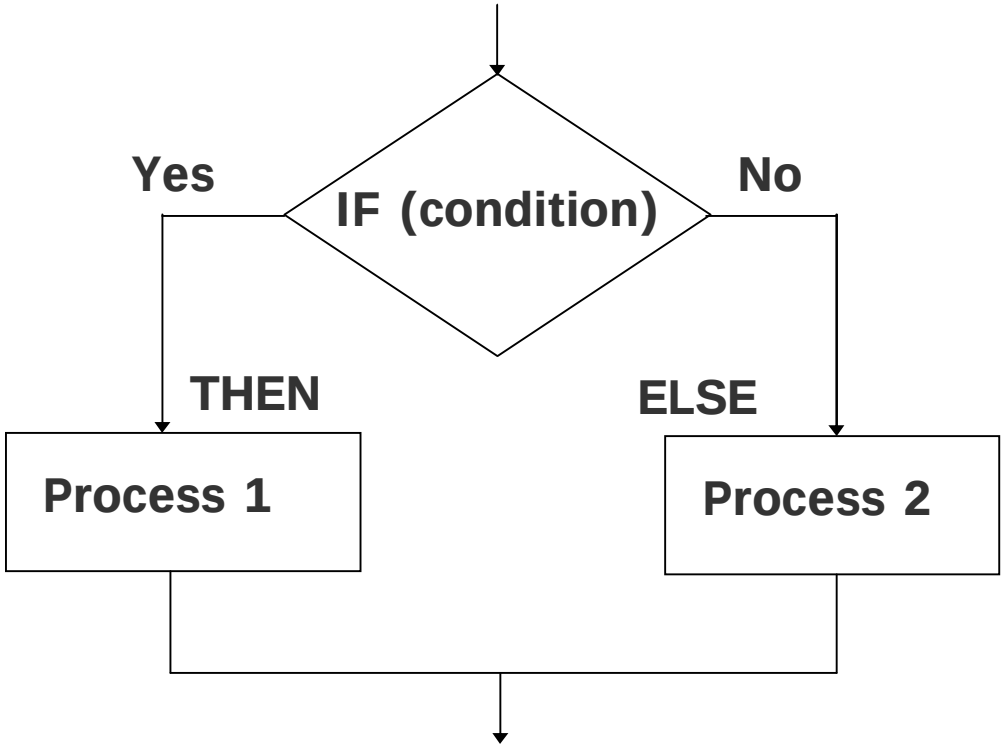
(b) Pseudocode

Selection Logic

- Also known as decision logic, it is used for making decisions

- Three popularly used selection logic structures are
  1. IF...THEN...ELSE
  2. IF...THEN
  3. CASE

Selection Logic (IF...THEN...ELSE Structure)



(a) Flowchart

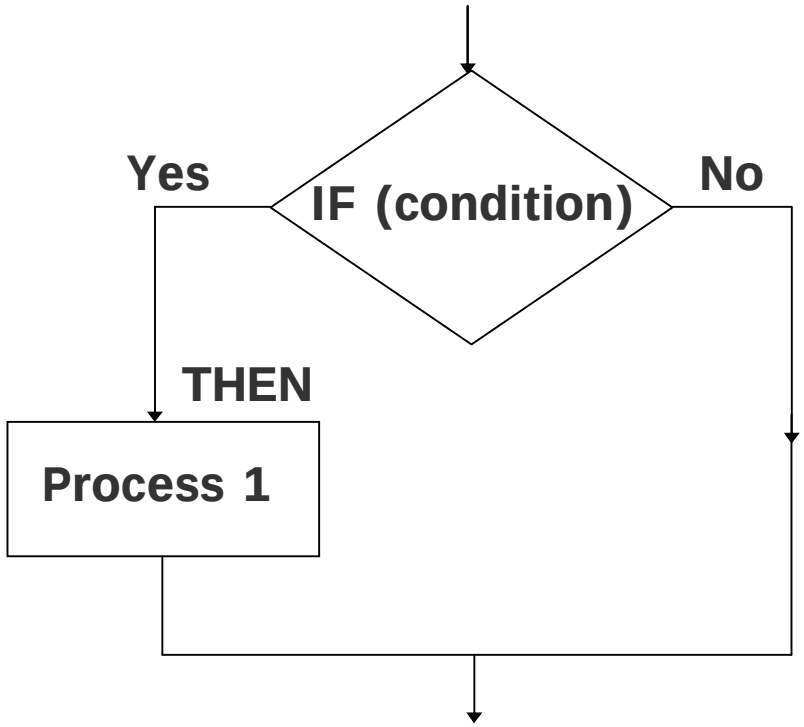
```
    .
    .
    .
IF Condition

    THEN    Process 1
    ELSE    Process 2

ENDIF
    .
    .
    .
```

(b) Pseudocode

Selection Logic (IF...THEN Structure)

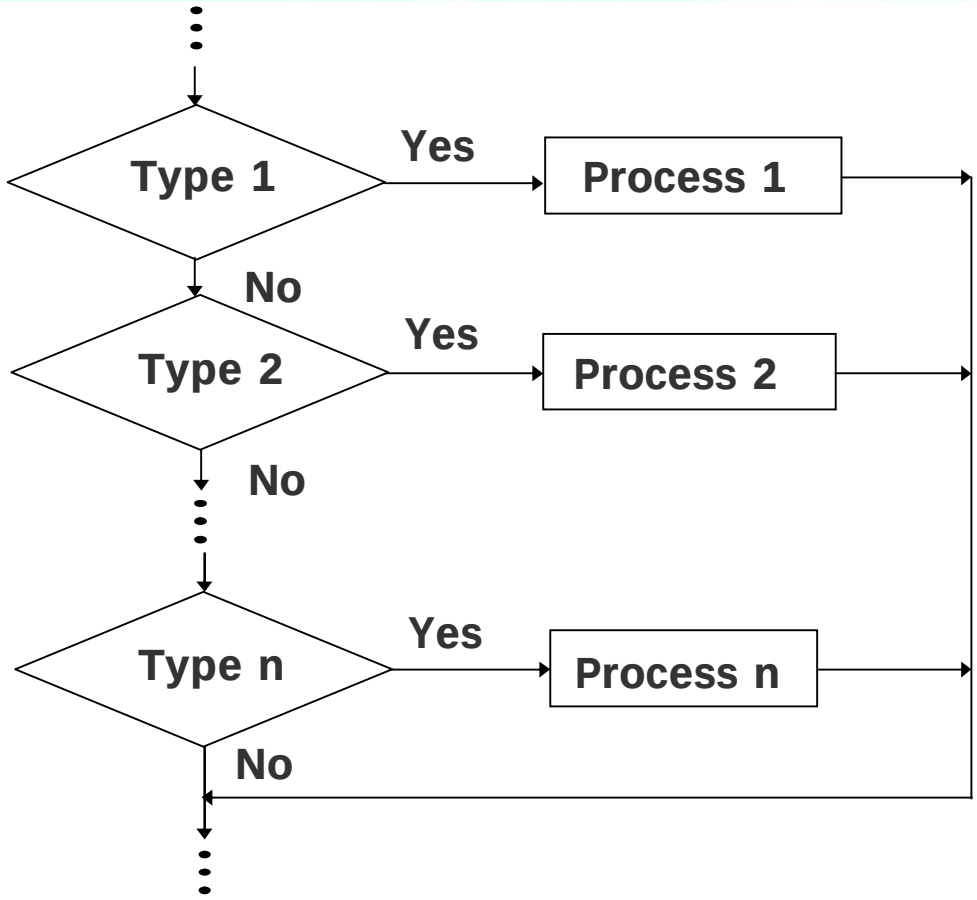


(a) Flowchart

```
:
IF Condition
    THEN    Process 1
ENDIF
:
```

(b) Pseudocode

Selection Logic (CASE Structure)



(a) Flowchart

```
CASE Type
    Case Type 1:    Process 1
    Case Type 2:    Process 2
    Case Type n:    Process n
ENDCASE
```

(b) Pseudocode

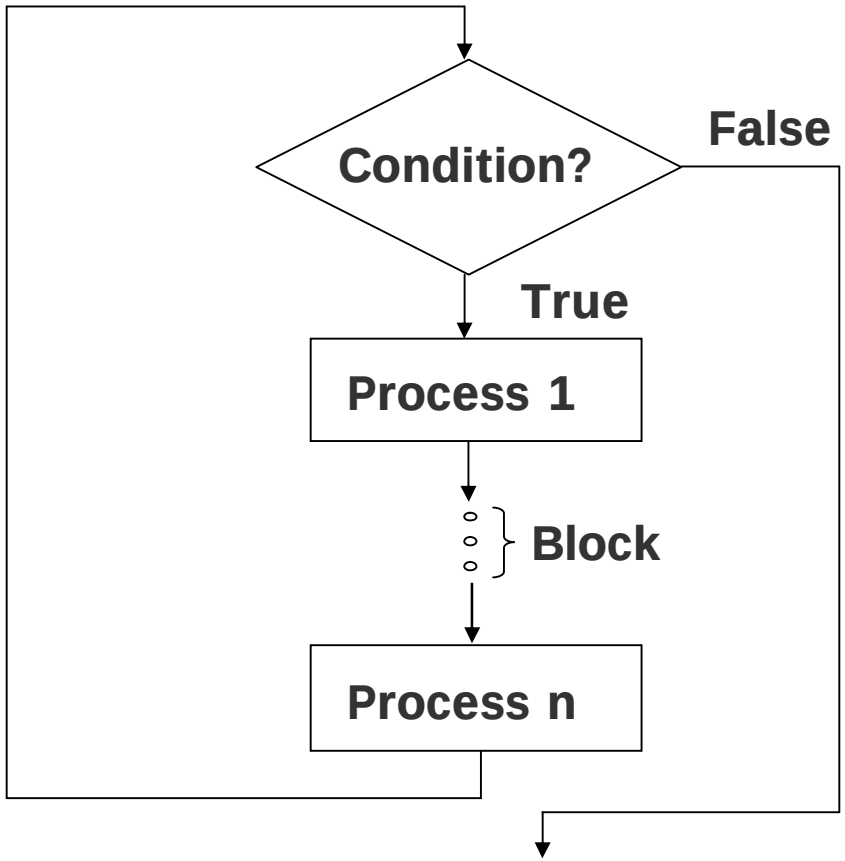
Iteration (or Looping) Logic

- Used to produce loops in program logic when one or more instructions may be executed several times depending on some conditions

- Two popularly used iteration logic structures are

  1. DO...WHILE
  2. REPEAT...UNTIL

Iteration (or Looping) Logic
(DO...WHILE Structure)

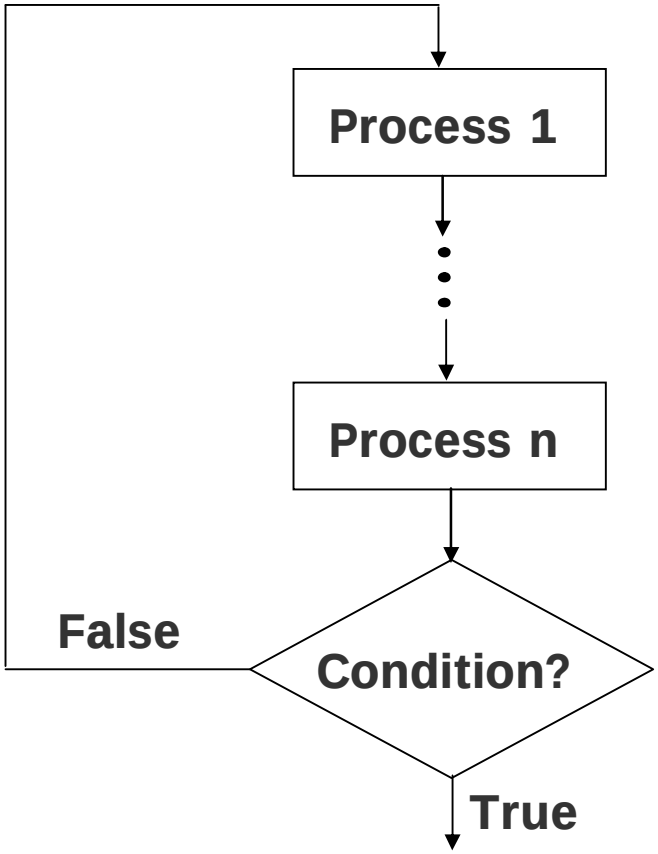


(a) Flowchart

```
:
DO WHILE Condition
    Process 1
    :
    Process n
ENDDO
:
```

(b) Pseudocode

Iteration (or Looping) Logic
(REPEAT...UNTIL Structure)



(a) Flowchart

```
    .
    .
    .
REPEAT
        Process 1
        .
        .
        .
        Process n
UNTIL Condition
    .
    .
    .
```

(b) Pseudocode

Sample Pseudocode (for Example 6)

```
Set Count to zero
Read first student record
DO WHILE Sexcode is not equal to Z
      IF Sexcode = F THEN
            Calculate Percentage
            IF Percentage = > 45 THEN
                  IF Percentage < 60 THEN
                        Write output data
                        Add 1 to Count
                  ENDIF
            ENDIF
      ENDIF
      Read next student record
ENDDO
Write Count
Stop
```

Advantages of Pseudocode

- Converting a pseudocode to a programming language is much more easier than converting a flowchart to a programming language
- As compared to a flowchart, it is easier to modify the pseudocode of a program logic when program modifications are necessary
- Writing of pseudocode involves much less time and effort than drawing an equivalent flowchart as it has only a few rules to follow

Limitations of Pseudocode

- In case of pseudocode, a graphic representation of program logic is not available
- There are no standard rules to follow in using pseudocode
- Different programmers use their own style of writing pseudocode and hence communication problem occurs due to lack of standardization
- For a beginner, it is more difficult to follow the logic of or write pseudocode, as compared to flowcharting

Key Words / Phrases

- Algorithm
- Basic logic structures
- Control structures
- Flowchart
- Iteration logic
- Looping logic
- Micro flowchart
- Macro flowchart
- Pseudocode
- Program Design Language (PDL)
- Sequence logic
- Selection logic
- Sentinel value
- Structured programming
- Trailer record