Virtual University

CS101-Introduction-to-Computing

MUHAMMAD IMRAN

MUHAMMAD IMRAN

Topic-001. Introduction to Computer Science

What is Computer Science?

Study of:

1. Hardware/Software
2. Programing
3. Networks
4. Graphics
5. Robots
6. Database
7. Security
8. Algorithmic Solutions
9. Information Processing

Applications:

- ✓ Telecom
- ✓ Banks
- ✓ Hospitals
- ✓ Software Houses
- ✓ Service Industry
- ✓ Pak Army
- ✓ Freelancing
- ✓ and many more

Job Web Portal in Pakistan

10 most demanded jobs:

1. Artificial Intelligence and Machine Learning
2. Data Science
3. Virtual Reality
4. IoT
5. Back-End Developer
6. Front-End Developer
7. UI Designer
8. Full-Stack Engineer
9. IT Manager
10. Quality Assurance Expert

Summary

What is computer Science

Why we should study CS:

1. Applications
2. Job Placement
3. Local market trend
4. International Market Trend
5. Most demanded jobs in CS

Topic-002. Breadth first learning

Covering abstract view of all major courses in CS

- ✓ Understanding what will be studied in CS
- ✓ Why each course is important
- ✓ Clarifying the bigger Picture

Topics:

- ✓ Search Engine Usage Techniques
- ✓ History of Computing
- ✓ Data Storage
- ✓ Data Manipulation
- ✓ Operating System
- ✓ Networking and the Internet
- ✓ Algorithms
- ✓ Programming Languages
- ✓ Scratch Tool
- ✓ Software Engineering
- ✓ Data Abstraction
- ✓ Database Systems
- ✓ Artificial Intelligence
- ✓ CS impact on society
- ✓ Content Filtering, Spam, International laws
- ✓ Word Processing
- ✓ Presentations Development
- ✓ Spreadsheet
- ✓ Spreadsheet
- ✓ Database MS Access
- ✓ Web Page Development

Summary

- ✓ Breadth First Learning
- ✓ Introduction to courses covered in CS
- ✓ What are the aims of each course
- ✓ What are the main topics in each course
- ✓ Which strengths each course builds in students
- ✓ What are possible paths of career in CS

Topic-003. Search engines

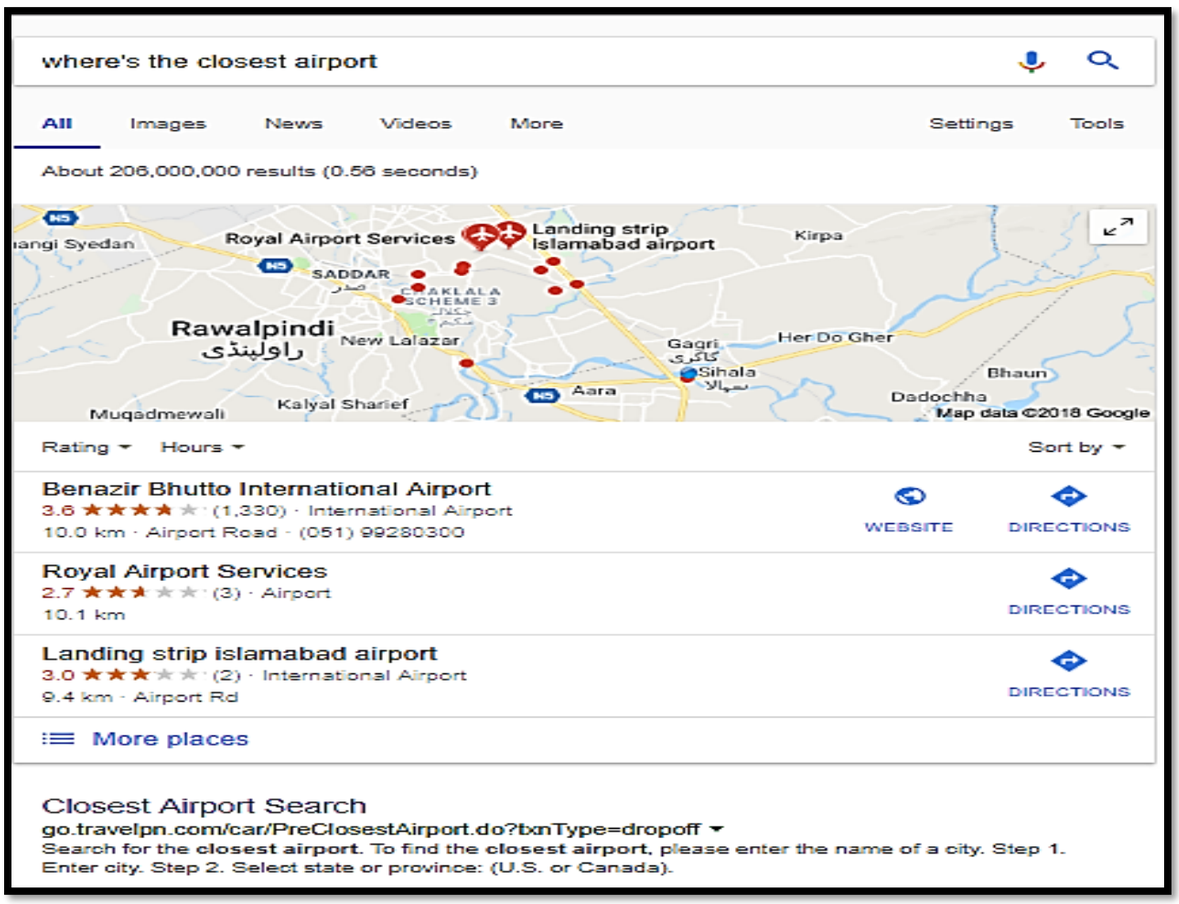
- ✓ Google
- ✓ Bing
- ✓ Yahoo

Google

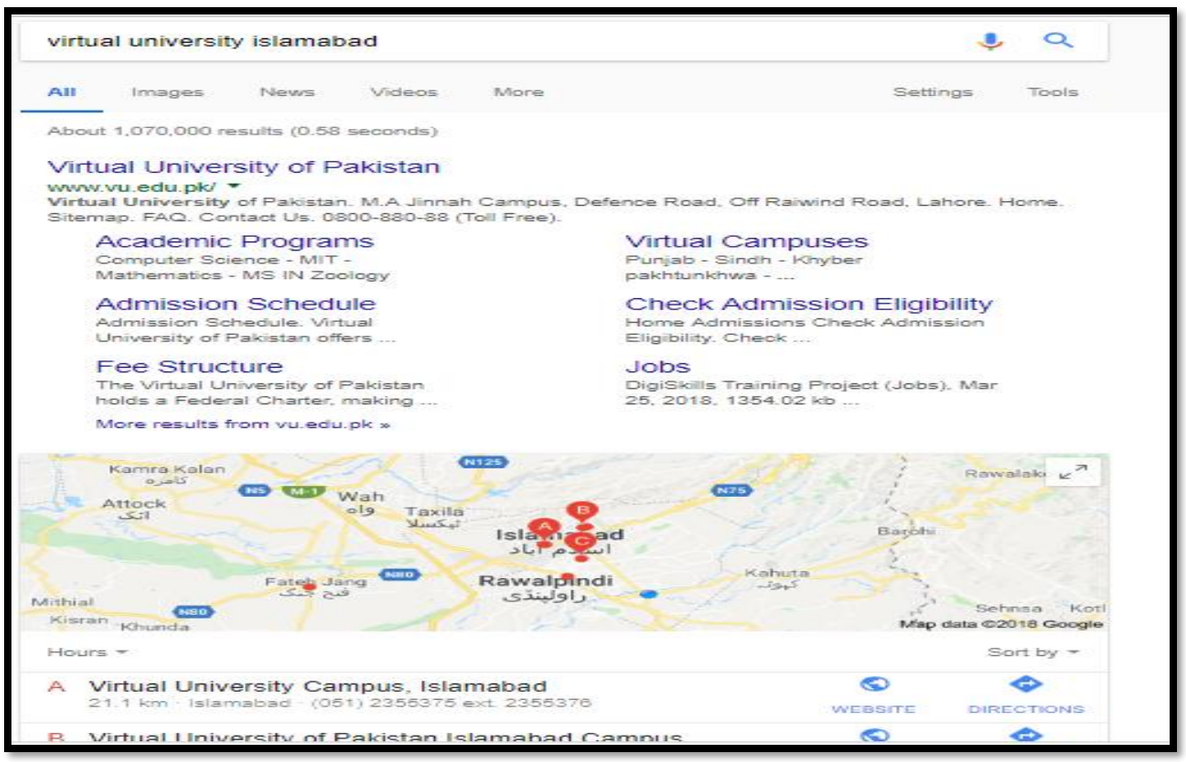
https://www.google.com/

Query Formulation

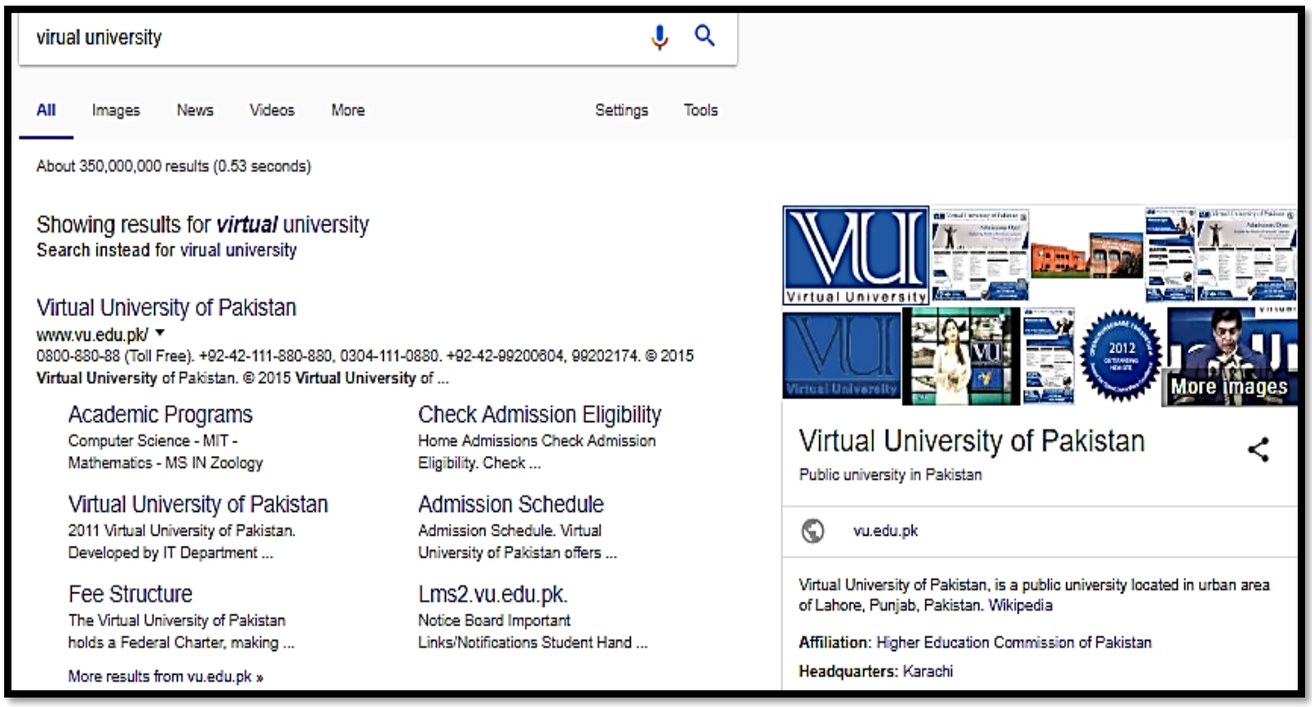
Start with simple terms



Specify location:

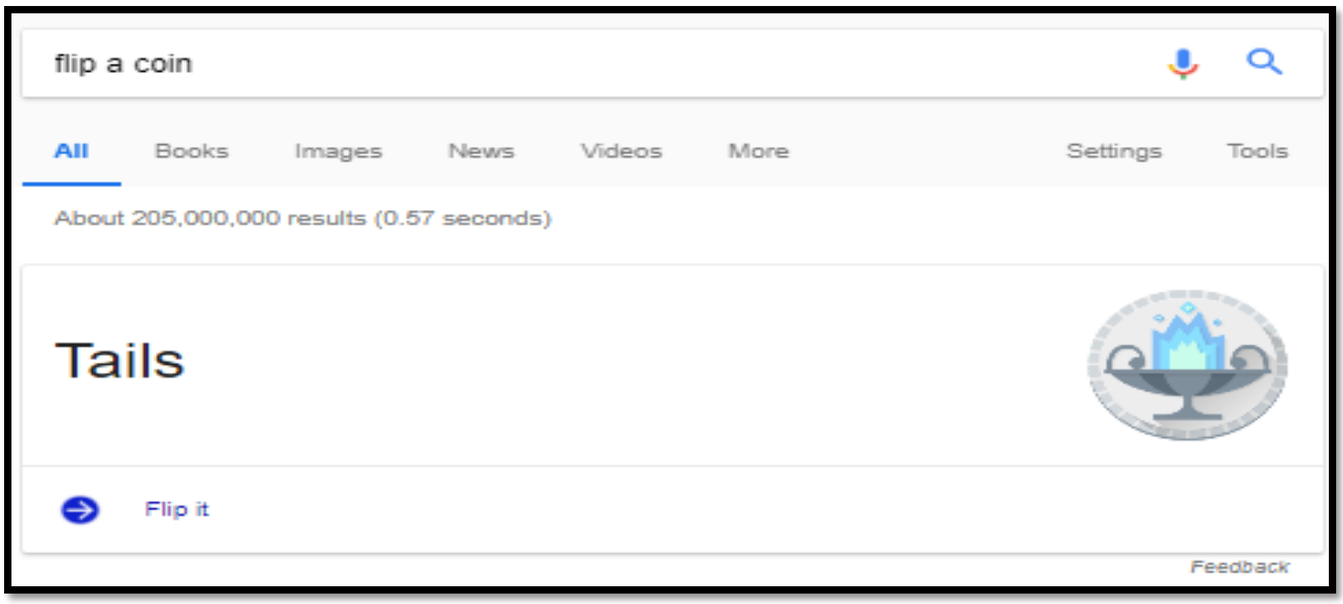


Spelling



Search using voice:

Flip a Coin:



Choose Words Carefully

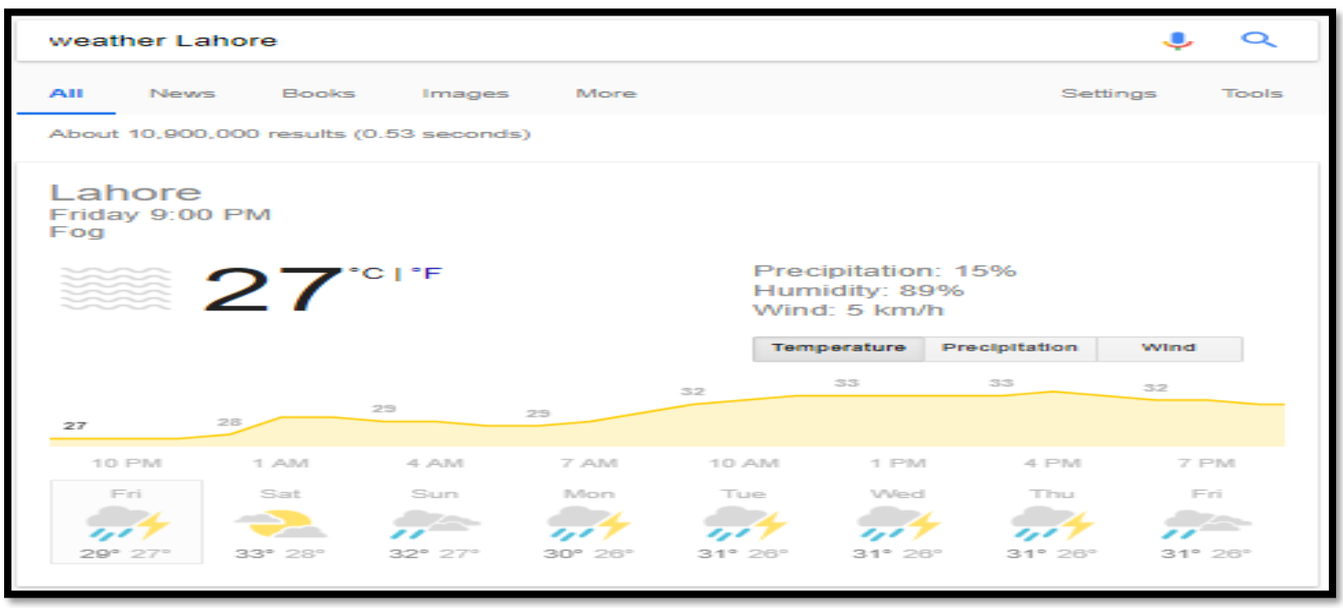
- ✓ Head hurts/headache
- ✓ head virtual university/Rector Virtual University
- ✓ Remote Education/Virtual education
- ✓ Apple could point to company and fruit

Capitalization

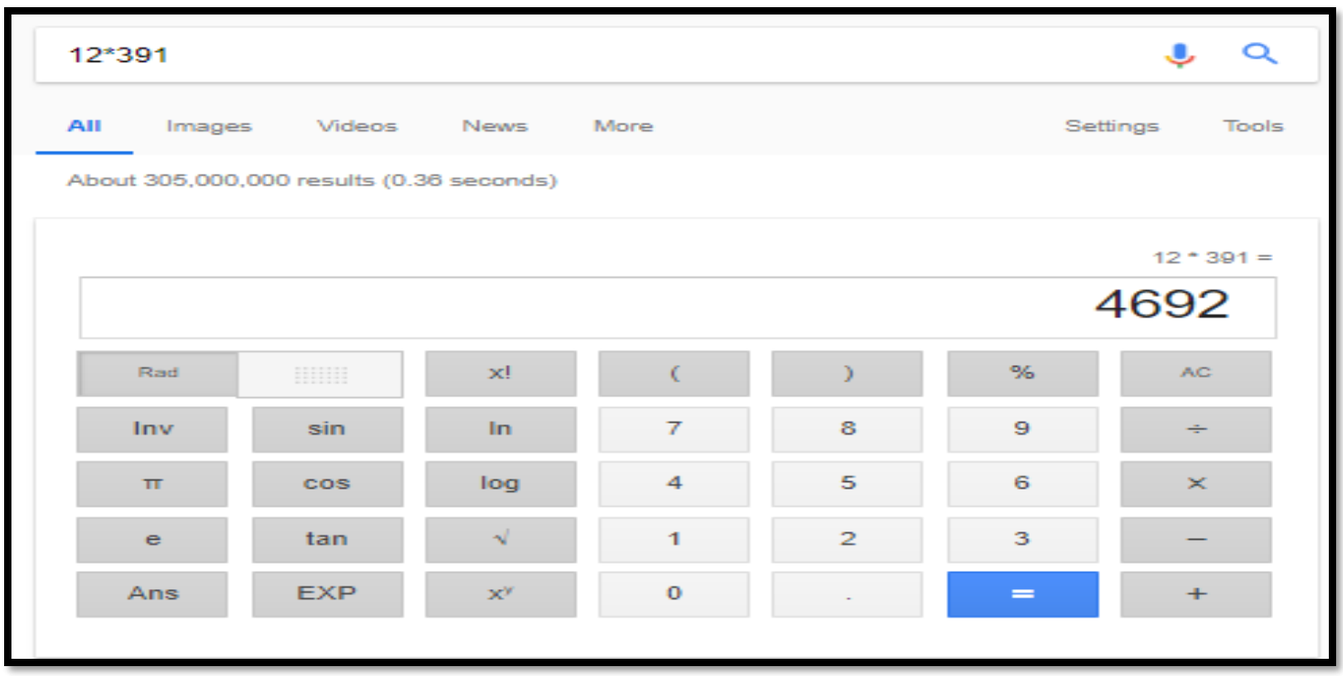
- ✓ **COMPUTER SCIENCE**
  VS
  computer science
  Summary
- ✓ Search Engines
- ✓ Google searching tips and effective query formulation

Topic-004. Searching tricks

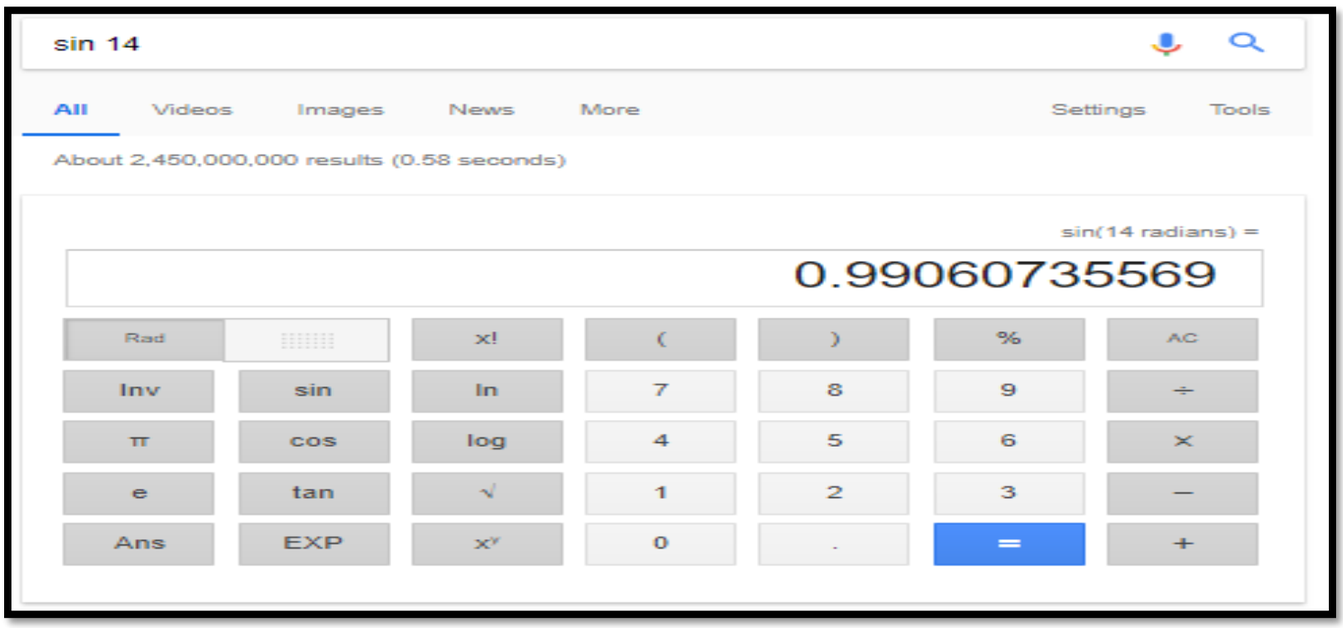
Weather Searching



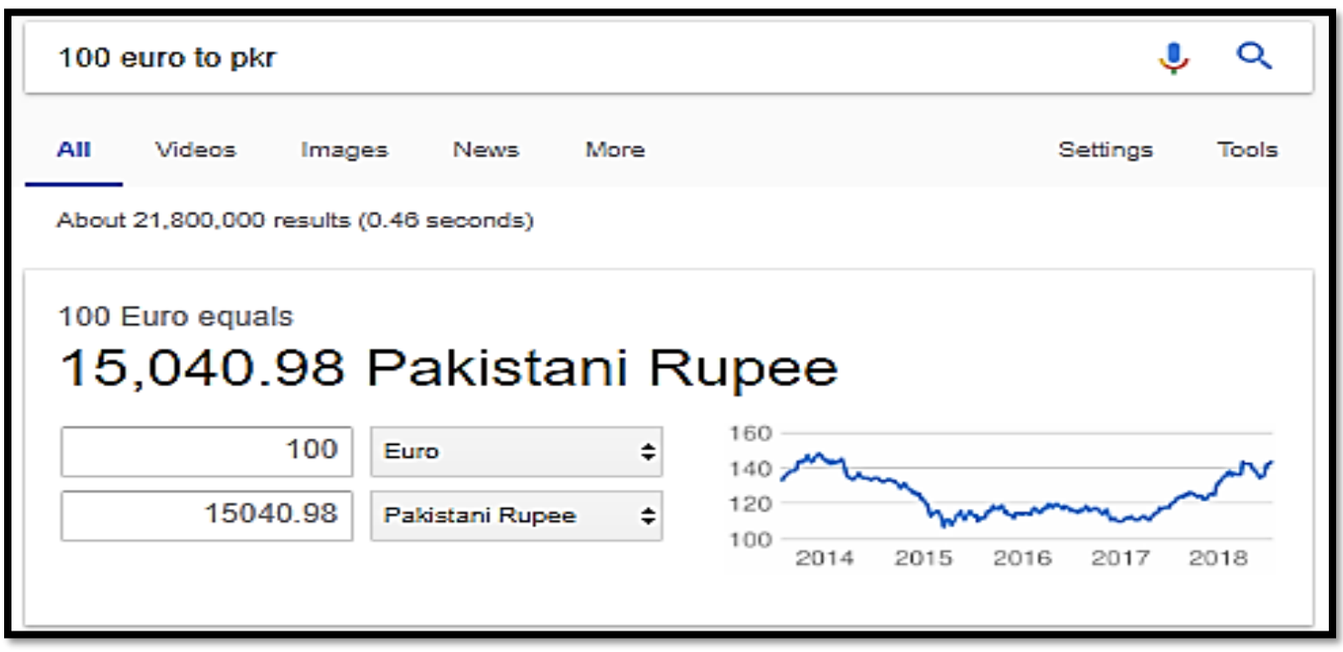
Calculations



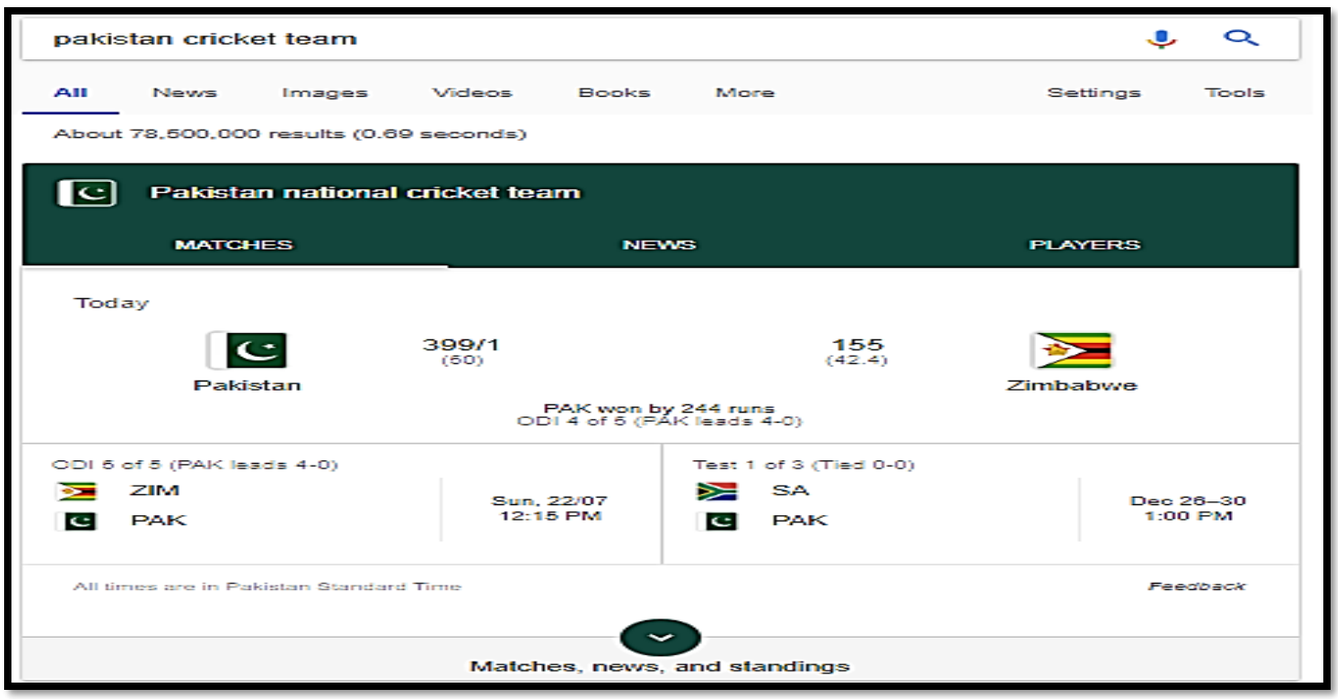
Calculations



Currency conversion



Sports

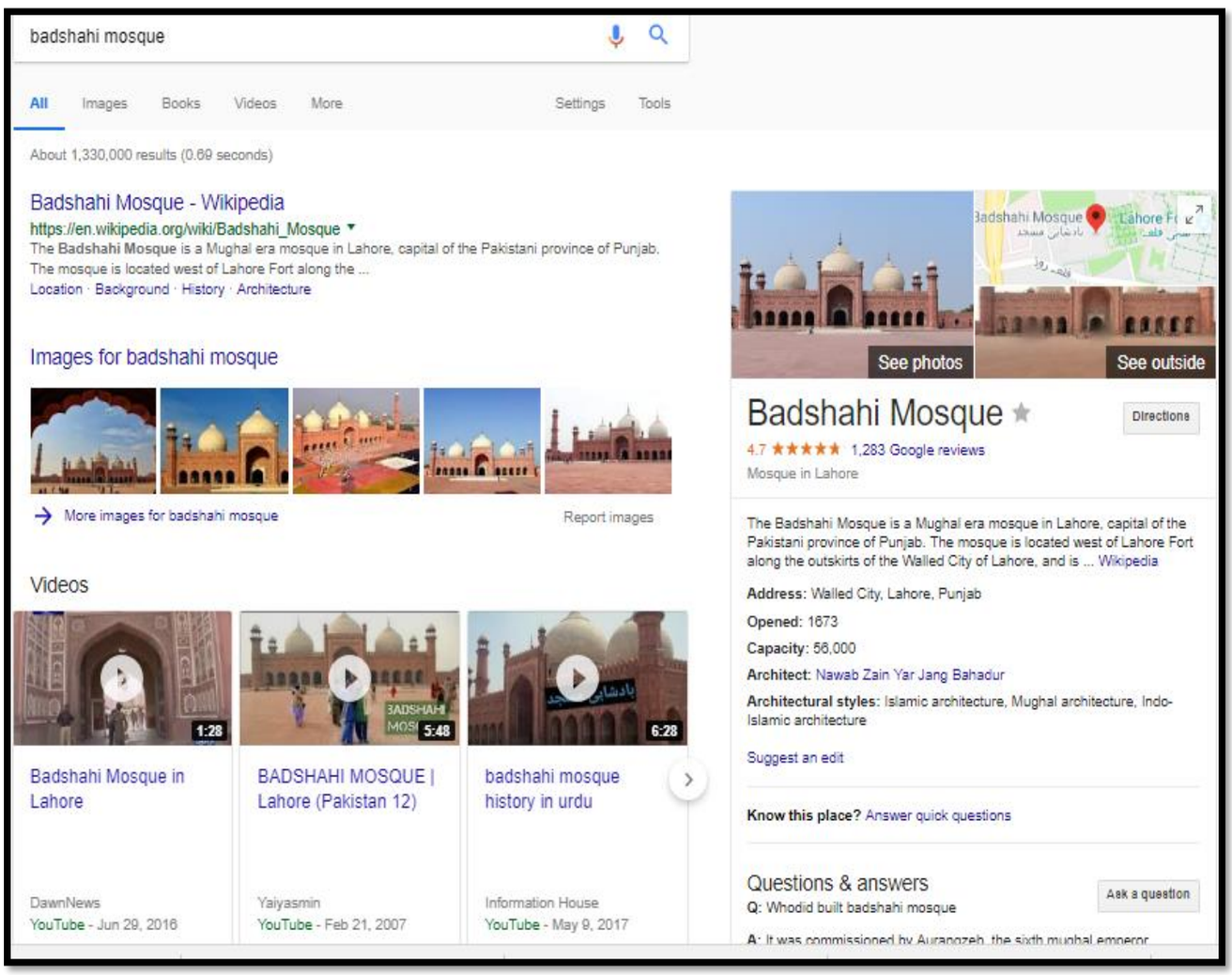


Quick Facts

- ✓ Celebrity
- ✓ Location
- ✓ Movie
- ✓ Song

Searching a Location

Badshahi Mosque



Summary

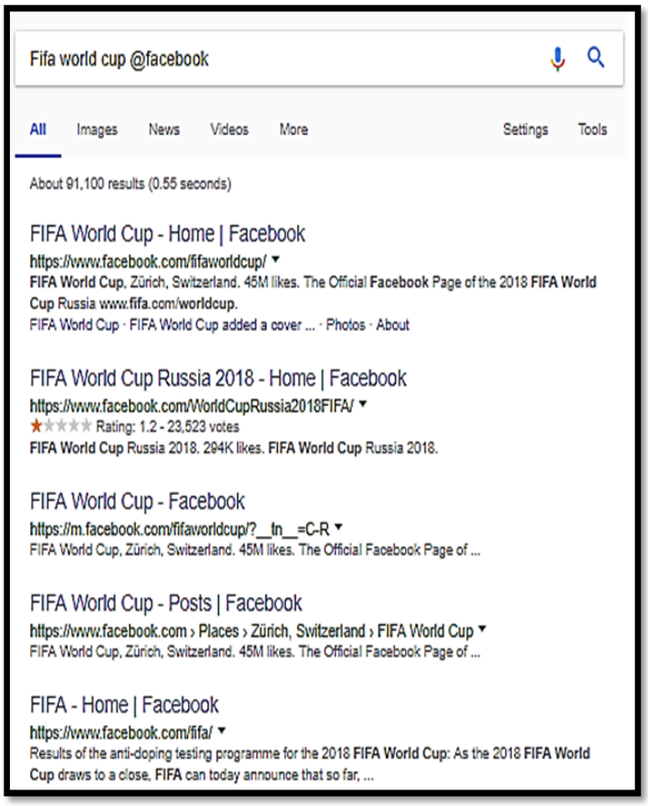
- ✓ Searching tricks
- ✓ Weather

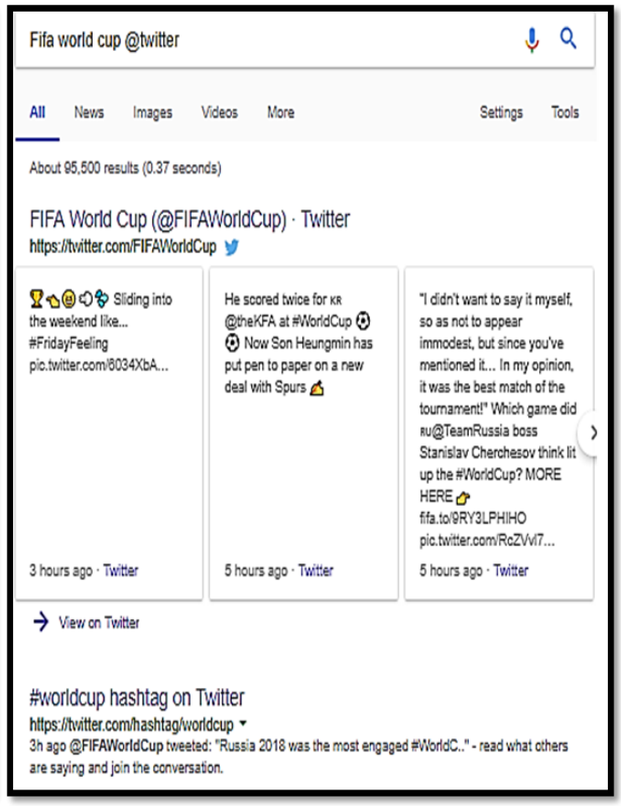
- ✓ Calculations
- ✓ Currency conversion
- ✓ Sports
- ✓ Location, celebrity

Topic-005. Search operators-01

Search Social Media

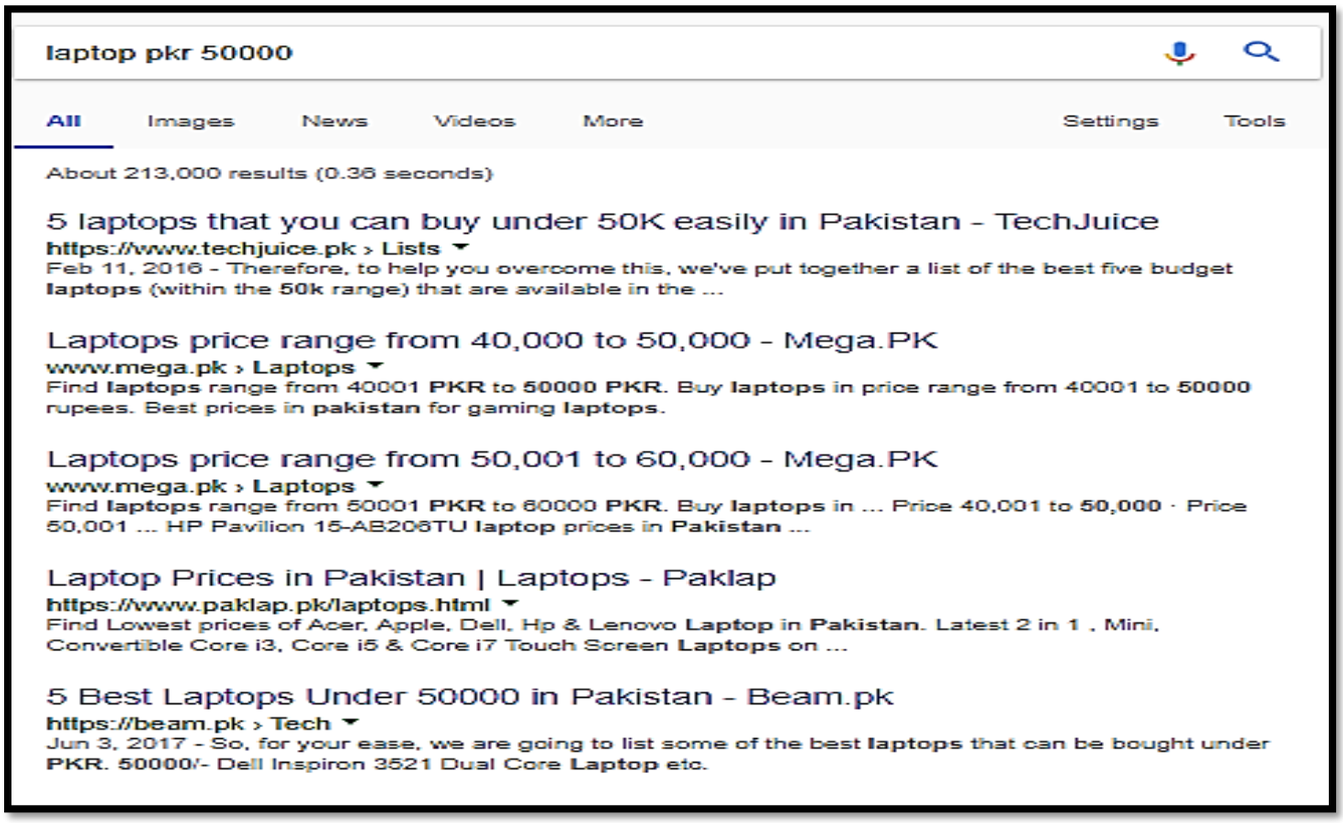
- ✓ Put @ in front of a word to search social media





Search for a price

- ✓ Put pkr in front of a number



Search Hashtags

- ✓ Put # in front of a word

- ✓ #education on Google

Exclude words from search

- ✓ Jaguar speed -car

Exact Match

- ✓ "tallest building in pakistan"

Wild Card

- ✓ “* is thicker than water”

Summary

Search operators

- ✓ @
- ✓ #
- ✓ pkr
- ✓ Exact match
- ✓ Exclude
- ✓ Wildcard

Topic-006. Search operators-02

Range of Numbers

- ✓ Pkr1000..pkr2000

Boolean Search

- ✓ AND
- ✓ OR (|)

Search for a specific Site

- ✓ virtual university site:youtube.com

Search for a related Site

- ✓ related:youtube.com

Info about website

- ✓ info:youtube.com

Cached version

- ✓ cache:youtube.com

Filetype

- ✓ “Virtual University” filetype:pdf
- ✓ “Virtual University” ext:pdf

Summary

Search operators

- ✓ Range numbers
- ✓ OR “|”
- ✓ Search within a site
- ✓ Search related sites
- ✓ Info site
- ✓ Cached version

Topic-007. Search operators-03

stock:
- ✓ stocks:aapl
- ✓ map:
- ✓ map:Lahore
- ✓ movie:
- ✓ movie:steve jobs
- ✓ Compare foods:
- ✓ http://www.twofoods.com/
- ✓ Define:
- ✓ Define:Computer
- ✓ Image search:
- ✓ https://images.google.com/
- ✓ Tilt:
- ✓ tilt

Summary

- ✓ Search operators
- ✓ Stock
- ✓ Map
- ✓ Movie
- ✓ Define
- ✓ Image search
- ✓ Tilt

Topic-008. Advanced search operators

intitle:

- ✓ Intitle:“iphone vs android”

allintitle:

- ✓ allintitle:iphone vs android

inurl

- ✓ inurl:2018 “virtual university”

allinurl

- ✓ allinurl:2018 “virtual university”

intext

- ✓ intext: “virtual university admissions 2018”

allintext

- ✓ allintext: virtual university admissions 2018

Proximity search

- ✓ AROUND (3)
- ✓ education AROUND(3) "virtual university"

Lets solve a complex problem

- ✓ We are interested to see how many of pages of a website are not secured?
- ✓ site:youtube.com –inurl:https

Summary

Advanced Search operators

- ✓ intitle
- ✓ inurl
- ✓ Intext
- ✓ Proximity search
- ✓ Complex problem

Topic-009. What you should not search on internet

Avoid Searching on Internet

Basis of Google Business:

- ✓ Personalized ads
- ✓ Based on your interests, location, searching history, viewing history, profile

Not to search

- ✓ Your email
- ✓ Medical issues and drugs
- ✓ Your name
- ✓ Your location
- ✓ Your favorite things

Dangerous to search

- ✓ pressure cooker bomb story
- ✓ Attacks
- ✓ Suicide bomb
- ✓ Killers/Underworld
- ✓ Terrifying insects
- ✓ Killing animals
- ✓ Poisons
- ✓ Murder
- ✓ Medical Symptoms
- ✓ How to make computer Virus
- ✓ Hacking
- ✓ Free Music

Sensitive information

- ✓ About Religion
- ✓ About Politics

Unpleasant Results

- ✓ Smokers Lungs
- ✓ Skin Condition

Summary

- ✓ What to Avoid while Searching
- ✓ Basics of search engine business
- ✓ Dangerous to Search
- ✓ Sensitive Information

Topic-010. Roots of computing

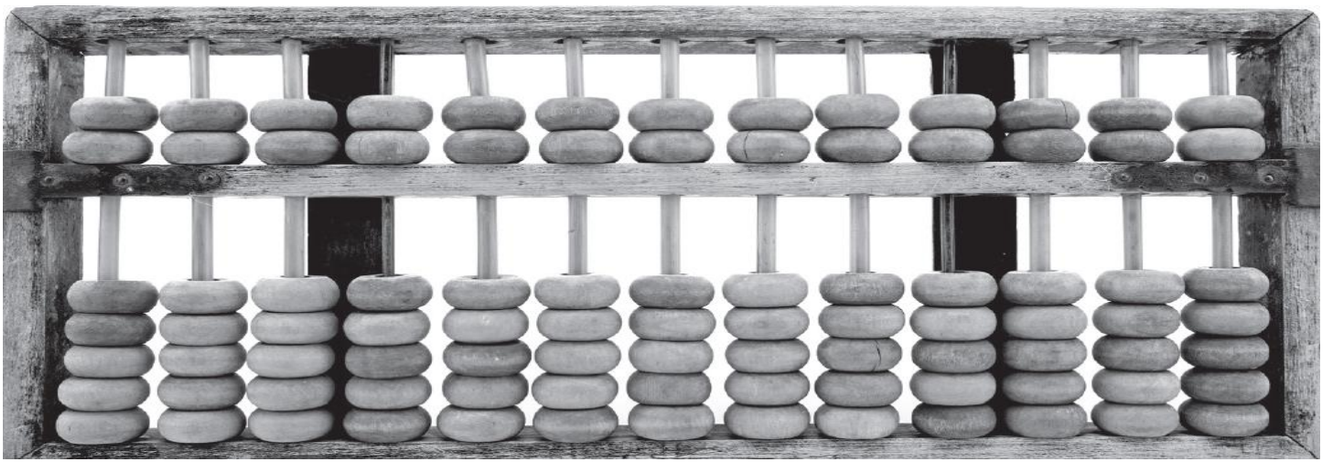
Roots of Computing

Mechanical vs Electronics-driven machines

Book: Computer Science : An Overview, 12th ed. By J. Glenn BrookShear and Dennis Brylow

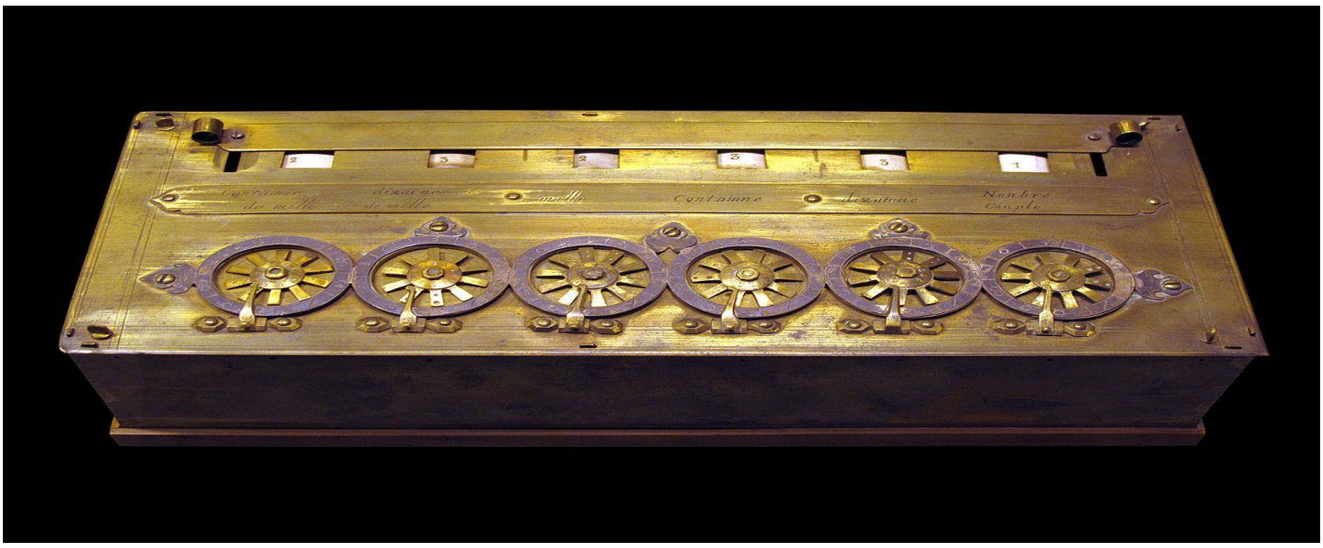
Abacus:

- ✓ Ancient China, Greek and Roman Civilization



Technology of Gears:

- ✓ Pascal from france (1623-1662)
- ✓ Leibniz (1646-1716) from Germany
- ✓ Babbage from England (1792-1871)



Data Storage

In Geared Technology:

- ✓ Pascal and Leibniz output in gear
- ✓ Babbage produced output on paper (punch cards)
- ✓ Punch card helped in algorithm changes

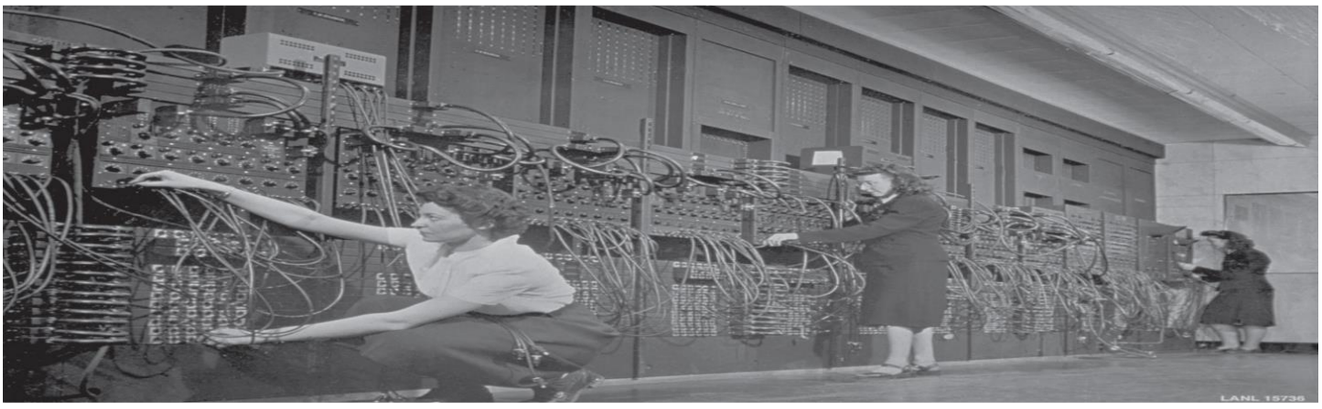
Mechanical to Electronic

Electronically Controlled

Mechanical Relays

- ✓ Cost effective production of geared Technology-challenge
- ✓ Electromechanical machine by Stibitz in 1940 at Bell Labs
- ✓ Mark-I at Harvard in 1944

Vacuum Tubes



ENIAC (electronic numerical integrator and calculator)

FactSheet: ENIAC by 1956

- ✓ Occupied 1800 square feet
- ✓ 20, 000 vacuum tubes
- ✓ 1500 relays
- ✓ 10, 000 capacitors
- ✓ 70, 000 registers
- ✓ 200 Kilo Watt electricity
- ✓ Weight 30tons
- ✓ Cost = $487, 000
- ✓ PKR = 62.5 millions

Rapid Advancement

- ✓ Transistor
- ✓ Integrated Circuits
- ✓ Size reduction
- ✓ Processing Power doubling in 2 years
- ✓ Desktop computer by Steve Jobs and Stephen Wozniak in 1976
- ✓ IBM launched PC in 1981
- ✓ Web
- ✓ Smart Phones

Topic-011. Bits

Basics:

- ✓ Information is coded as pattern of 0 or 1
- ✓ Short form of Binary Digits
- ✓ One bit can contain only one value 0 or 1

What they can represent:

- ✓ Representing numbers, text, audio, video, image etc
- ✓ In Chip electric Charge 0/1

Different Units

1 Bytes	8 Bits
1 Kilo Byte	1024 Bytes
1 Mega Byte	1024 Kilo Bytes
1 Giga Byte	1024 Mega Byte
1 Tera Byte	1024 Giga Byte

Patterns using Bits

- ✓ 1 bit - 2 patterns
- ✓ 2 bits - 4
- ✓ 3 bits - 8
- ✓ 4 bits - 16
- ✓ 5 bits - 32
- ✓ 6 bits - 64
- ✓ 7 bits - 128
- ✓ 8 bits - 256 - one byte

Bytes and Characters

- ✓ A = 65
- ✓ B = 66
- ✓ a = 97
- ✓ Space = 32

Decimal to Binary

- ✓ Lets represent 25 in binary

128	64	32	16	8	4	2	1
0	0	0	1	1	0	0	1

Summary

Bits and Bytes

- ✓ Basics
- ✓ Binary Representation
- ✓ Units
- ✓ Patterns
- ✓ How to represent a decimal value in Binary

Topic-012. Boolean operations

George Boole (1815-1864)

Imagine:

- ✓ O means False
- ✓ 1 means True
- ✓ AND
- ✓ OR
- ✓ XOR
- ✓ NOT
- ✓ Recall AND/OR usage in search engines
- ✓ AND - OR Operations

AND

Inputs — Output

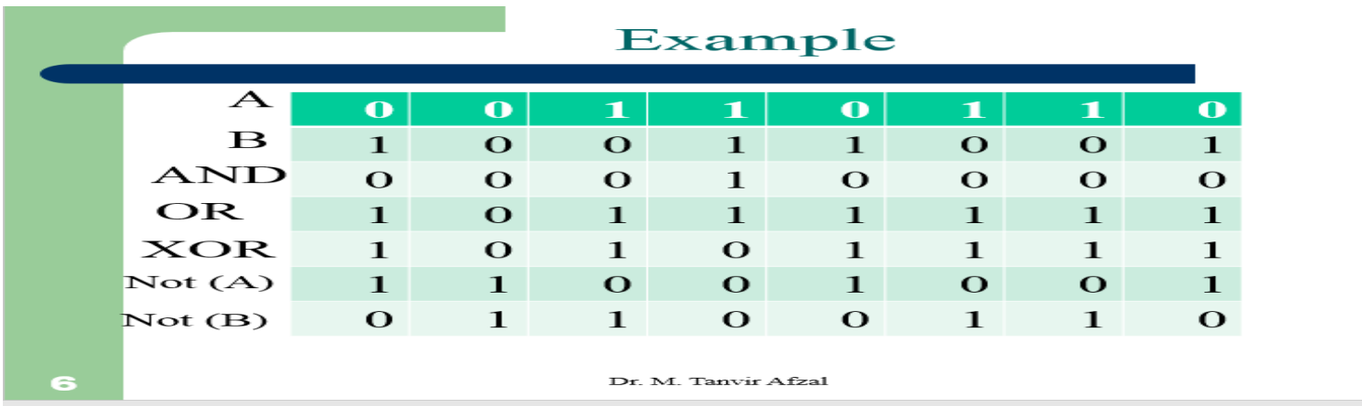
Inputs	Output
0 0	0
0 1	0
1 0	0
1 1	1

OR

Inputs — Output

Inputs	Output
0 0	0
0 1	1
1 0	1
1 1	1

Example



Summary

Boolean Operators

- ✓ AND
- ✓ OR
- ✓ XOR
- ✓ NOT
- ✓ Gates

Topic-013. Hexadecimal notation

Need:

- ✓ Bit patterns are large –Streams
- ✓ Understanding large sequence is difficult for humans
- ✓ We use Shorthand notation

Representation Motivation

- ✓ It represents 4 bits as single symbol
- ✓ Motivation: computer sequence is multiple of 4's

Representation

Bit pattern	Hexadecimal representation
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	A
1011	B
1100	C
1101	D
1110	E
1111	F

Lets solve some real problems

Q#5 (b) on page 38
Binary to Hexadecimal
1110100001010100010111
1110 1000 0101 0101 0001 0111

Q#6 (d) on page 38
Hexadecimal to Binary
0100

Dr. M. Tanvir Afzal

Bit pattern	Hexadecimal representation
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	A
1011	B
1100	C
1101	D
1110	E
1111	F

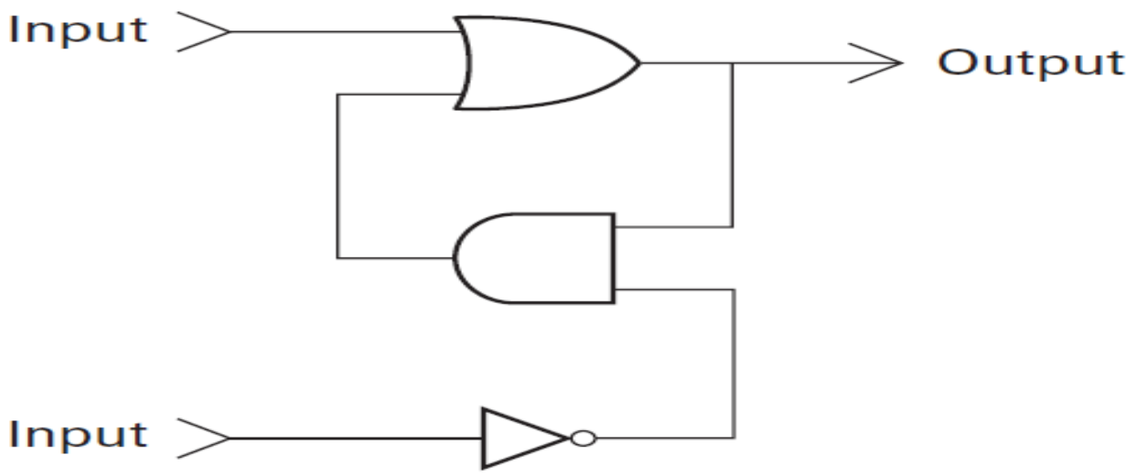
Summary

Hexadecimal Notation

- ✓ Motivation
- ✓ Why 4 bits
- ✓ Representation
- ✓ Examples

Topic-014. Storing a bit

Flip-Flop



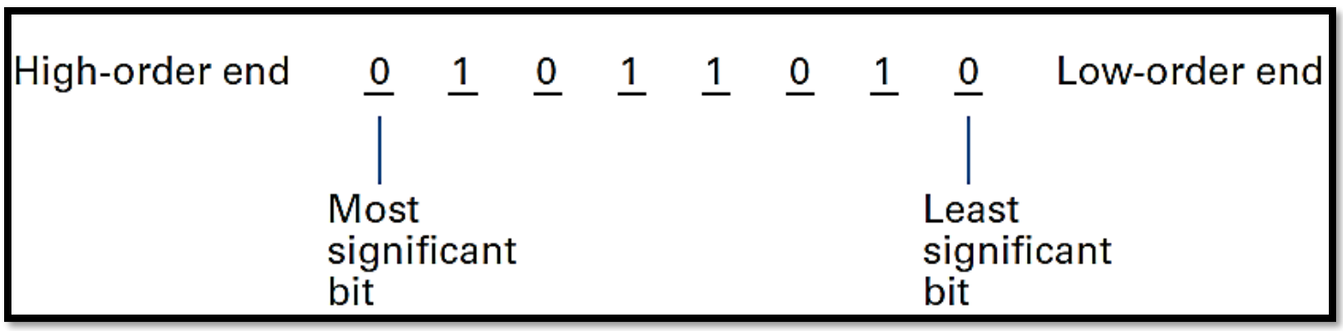
Flip-flop characteristics

- ✓ If both inputs are 0, then it retains its value.
- ✓ If upper bit is changed to 1, stored value changes to 1
- ✓ If lower bit is changed, stored value changes to 0

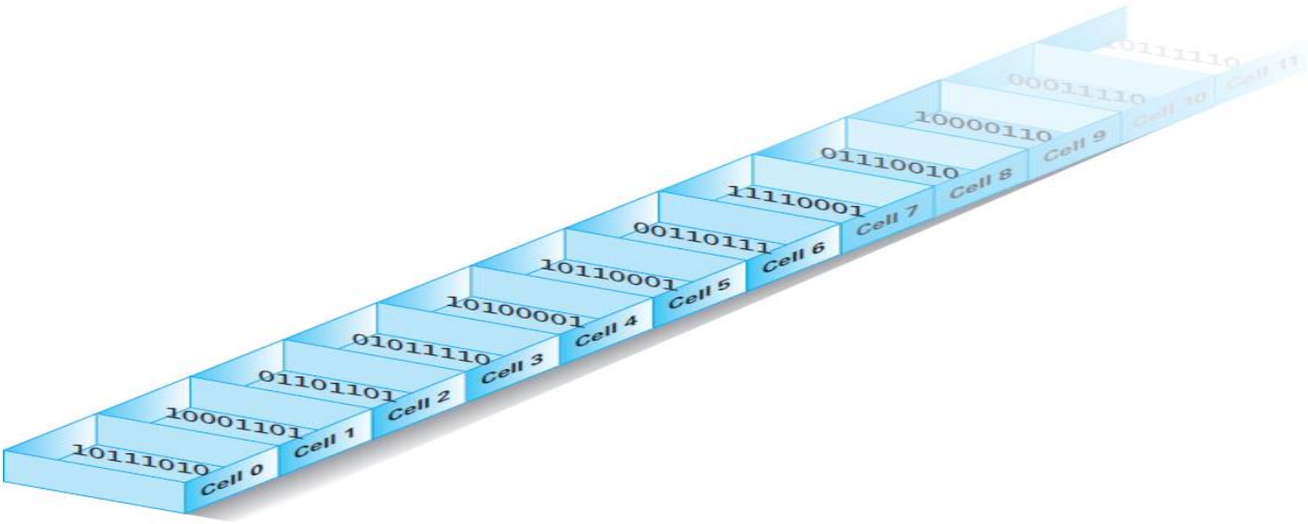
Main memory Organization

- ✓ Manageable units Cells – 8 bits
- ✓ Household devices have few hundred of cells
- ✓ Large computer may have billions of cells

Byte size Organization



Cell Address



Main memory

- ✓ Other circuits can store/retrieve data at any address – RAM
- ✓ Stores bits as tiny electric Charge, refreshes many times a second- DRAM
- ✓ Reduced time to retrieve content from cell – SDRAM

Summary

Main Memory:

- ✓ Flip-flop
- ✓ Memory Organization
- ✓ Cell Address
- ✓ RAM/DRAM/SDRAM

Topic-015. Magnetic systems

Why we need Mass Storage:

- ✓ Non-Volatility
- ✓ Large storage space
- ✓ Low cost
- ✓ Ability to remove for Archival purpose

Available Mass Storage:

- ✓ Magnetic Disks
- ✓ CDs
- ✓ DVDs
- ✓ Magnetic Tapes
- ✓ Flash Drives etc

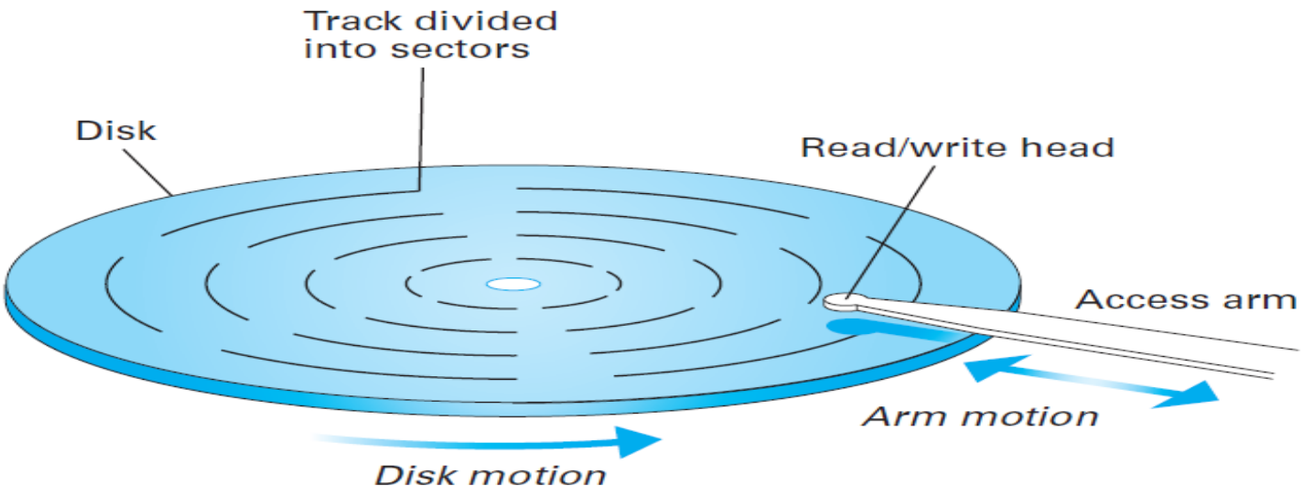
Magnetic Systems

- ✓ Dominated Mass Storage Arena
- ✓ Today: Magnetic Dis and Hard disk Drive

How It Works

- ✓ Thin spinning disk with magnetic coating to hold data
- ✓ Read/Write Heads placed above or below disk

- ✓ Each head traverses a circle called track



Tracks

- ✓ Normally, Each track is divided into equal sectors
- ✓ Sectors have equal number of bits: 512 bytes to few KBs.
- ✓ Outer tracks contain more information.

Zoned-bit recording

- ✓ Adjacent tracks form Zones.
- ✓ A typical disk contains 10 zones
- ✓ All tracks within a zone have equal number of sectors.

Evaluation of Disk Performance

Parameters

- ✓ Seek Time
- ✓ Rotation Delay/Latency Time
- ✓ Access Time
- ✓ Transfer Rate

Seek Time

- ✓ Time required to move the read/write heads from one track to another

Rotation Delay

- ✓ Average amount of time required for the desired data to rotate around to the read/write head once the head has been positioned over the desired track

Access Time

- ✓ the sum of seek time and rotation delay

Transfer rate

- ✓ the rate at which data can be transferred to or from the disk

Summary

Magnetic System

- ✓ Mass storage to Magnetic Systems, their needs
- ✓ How it works
- ✓ Evaluation parameters

Topic-016. Optical systems

Examples

- ✓ Compact Disk (CDs)
- ✓ Digital Versatile Disks **(**DVDs)
- ✓ Blu-ray Disks (BDs)

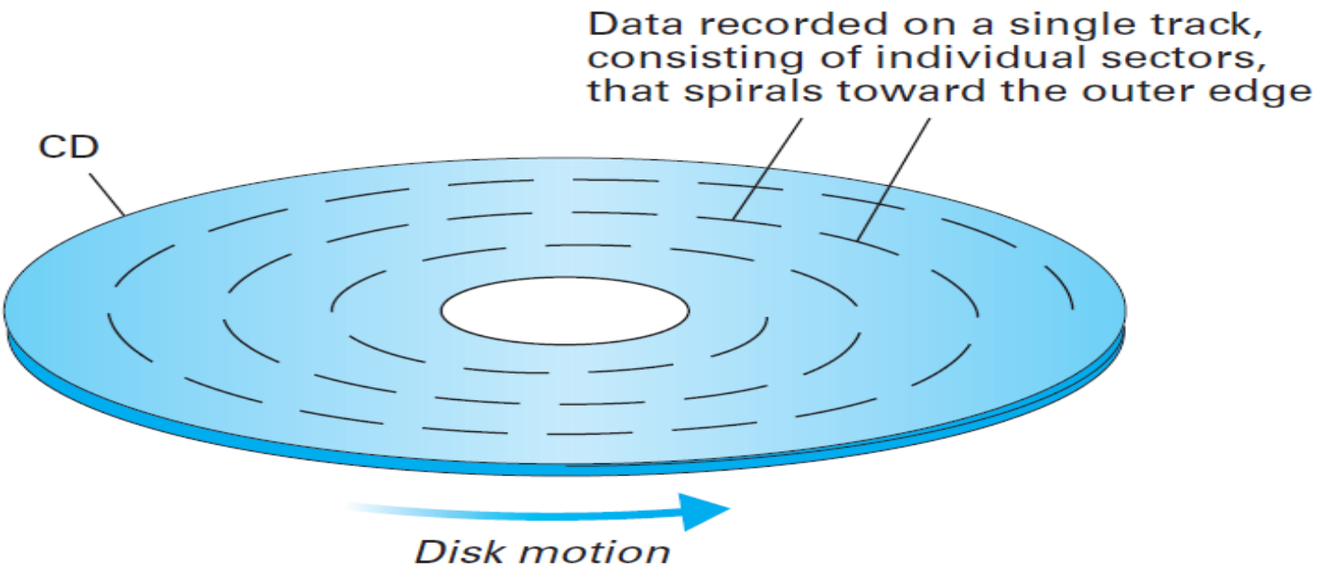
Compact Disk

- ✓ 12 centimeter in diameter
- ✓ Consists of reflective material

Data Storage on CD

- ✓ Information is recorded on them by creating variations in their reflective surfaces.
- ✓ Data is stored on a single track spiral from inside-outside.
- ✓ Each track is divided into sectors,
- ✓ capacity of a sector is 2KB of data, 1/75 of a second audio music.

CD



Data Retrieval on CD

- ✓ Retrieval using a Laser that detects the irregularities on the reflective surface of the CD as it spins

CD Technology

- ✓ Initially applied on audio recording using format CD-DA (compact disk-digital audio)
- ✓ Format is being used even today to store computer data
- ✓ Single Track

Random Retrieval

As all sectors are not individually accessible in Optical system having one track, therefore, the data retrieval is faster in magnetic systems than optical systems.Optical System is best suited for long continuous string of data.

DVDs (Digital Versatile Disks)

CDs capacity is 600 to 700 MB, however, DVD has space in GBs as have multiple semi-transparent layers, which can be viewed by precise focused laser.

BDs (Blue Ray Disks)

BDs use a blue-violet spectrum of light (instead of red), which is able to focus its laser beam with very fine precision. BDs provide five time capacity than DVDs.

Summary

Optical System

- ✓ CD, DVD, BDs,
- ✓ Data Storage and Retrieval
- ✓ Continuous access over random access

Topic-017. Flash drives

Flash **Drives**

Issues in Magnetic and Optical Systems

- ✓ Physical motion is required
- ✓ Moving Read/Write heads
- ✓ Aiming Laser beams
- ✓ All this takes lot of time in comparison with electronic circuitry
- ✓ Bits are stored by sending electronic signals directly to the storage medium where they cause electrons to be trapped in tiny chambers of silicon dioxide
- ✓ Chambers are able to store data for years without external power
- ✓ Repeated erasing slowly damages the silicon dioxide chambers
- ✓ Not suitable in place of Main memory
- ✓ Suitable where alterations can be controlled like cameras, smart phones, portable devices
- ✓ Not reliable as optical disks for long term

SSDs (Solid-State Disks)

- ✓ Larger Flash memory devices designed to take place of magnetic disks
- ✓ Quite operations and low access time
- ✓ However, costly than magnetic systems

SD (Secure Digital Memory Cards)

SD Cards

- ✓ Provide up to few GBs of storage
- ✓ Available in Smaller, mini, and Micro sizes
- ✓ SDHC (High Capacity) can provide 32 GBs
- ✓ SDXC (Extendable Capacity) may exceed to TB.
- ✓ Compact physical size, suitable for car navigation, cameras etc

Summary

Flash Memories

- ✓ Flash drives, SSDs, SDs, SDHC, SDXC
- ✓ No physical motion
- ✓ Suitable for portable devices

Topic-018. Representing text

Representation as Code

- ✓ Each Textual Symbol is represented with a unique bit pattern
- ✓ Normally 8 bits for each character
- ✓ “Virtual University”
- ✓ 18 characters = 18*8 (144) bits or 18 byte

Codes

- ✓ In 1940’s and 1950’s many such codes were designed
- ✓ American National Standards Institute (ANSI)
- ✓ American Standard Code for Information Interchange (ASCII

ASCII

- ✓ 7 bit for information and most significant bit has zero
- ✓ 27combinations= 128
- ✓ Uppercase, lower case, punctuation marks, digits 0 to 9, line feed, carriage returns, and tabs

ASCII codes

- ✓ Go to 577 of your book

01001000	01100101	01101100	01101100	01101111	00101110
H	e	l	l	o	.

Limitations of ASCII codes

- ✓ Only 128 characters
- ✓ International Organization for Standardization (ISO) come-up with many extensions to ASCII
- ✓ One to support western language symbols.

Two Issues

- ✓ 256 are still insufficient to denote all language symbols
- ✓ Document having multiple languages could not be read as it should follow a one standard

Unicode

- ✓ Internationalization of codes by manufacturers of hardware and software
- ✓ Unique patterns of 21 bits
- ✓ Compliance with ASCII
- ✓ Supporting thousands of character sets of Chinese, Herbew, Japaneze,

UTF-8

- ✓ Uses 24 to 32 bits having large possibilities for expansion
- ✓ 224= 16,777,216 unique symbols
- ✓ File consisting of long symbols encoded with ASCII or Unicode is called text file.

Summary

Representing Text

- ✓ How text is represented in binary
- ✓ ASCII and Unicode

Topic-019. Representing numeric values

Issues in storing numeric as Unicode

- ✓ Inefficient suppose you want to store 12, you would need 16 bits to do that
- ✓ 99 could be stored in 16 bits
- ✓ We will learn 16 bits can store 65,535 numeric values

Binary Notation

- ✓ Using only digits 0 and 1.
- ✓ Lets discuss an example of representing numeric values using binary notation
  Example 3 bits (counting 0 to 7)

Numeric Value	Binary
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

Adding one more bit

Numeric	Binary
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111

Numeric	Binary
8	1000
9	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111

Numeric Values Storage

Binary Notation Variations

- ✓ Two's complement for storing whole numbers
- ✓ Floating point notation for fractional numbers

Summary

Storing Numeric Values

- ✓ Issues in Unicode for storing numeric values
- ✓ Binary notation

Topic-020. Representing images

Pixel

- ✓ Collection of dots – Pixel short for Picture Element.
- ✓ Appearance of each pixel is encoded to form bit map.
- ✓ Many display devices, printers work on Pixel concept.

Encoding Methods

Pixel to bit map

- ✓ In black and white images, each pixel is represented as one bit – e.g. 0 for black and 1 for white
- ✓ Often used in Facsimile

Handling shades

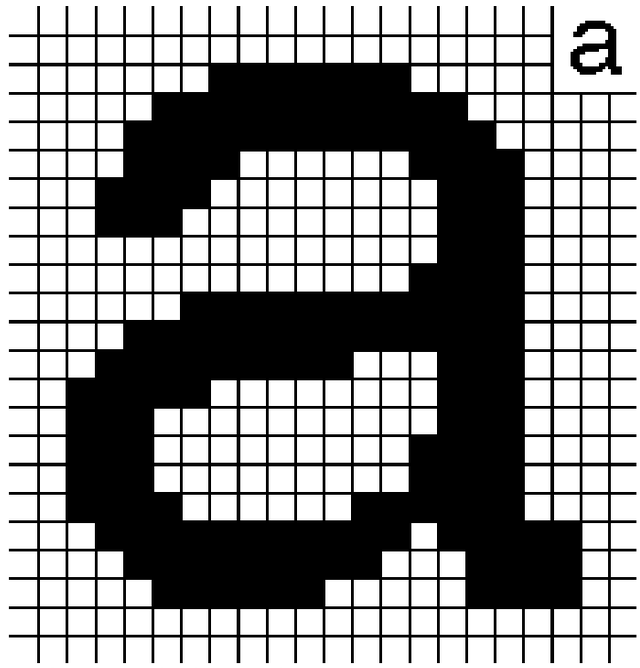
- ✓ 8 bits
- ✓ Variety of shades of grayness.

Colorful Images

- ✓ RGB encoding
- ✓ One byte for Red, Green, and Blue
- ✓ Brightness Chrominance

Brightness Chrominance

- ✓ One brightness component and two color components.
- ✓ Brightness component= Pixel's luminance (sum of red, green, and blue)
- ✓ Blue Chrominance and Red Chrominance
- ✓ Difference between luminance and amount of blue or red.



Bit map issues

Scaling of Images

- ✓ Scaling to a larger size needs more pixels
- ✓ Digital Zoom

Alternative Method

Geometric Structures

- ✓ Collection of lines and curves
- ✓ Using Analytical Geometry
- ✓ Technique: How geometric structures should be displayed rather Pixel reproduction

Geometric Structures

Scalable Fonts

- ✓ TrueType by Microsoft and Apple
- ✓ PostScript by Adobe
- ✓ Also popular in Computer Aided Design (CAD)

Summary

Representing Images

- ✓ Pixel
- ✓ Bit map
- ✓ Brightness Chrominance
- ✓ Geometric Structures

Topic-021. Representing sound

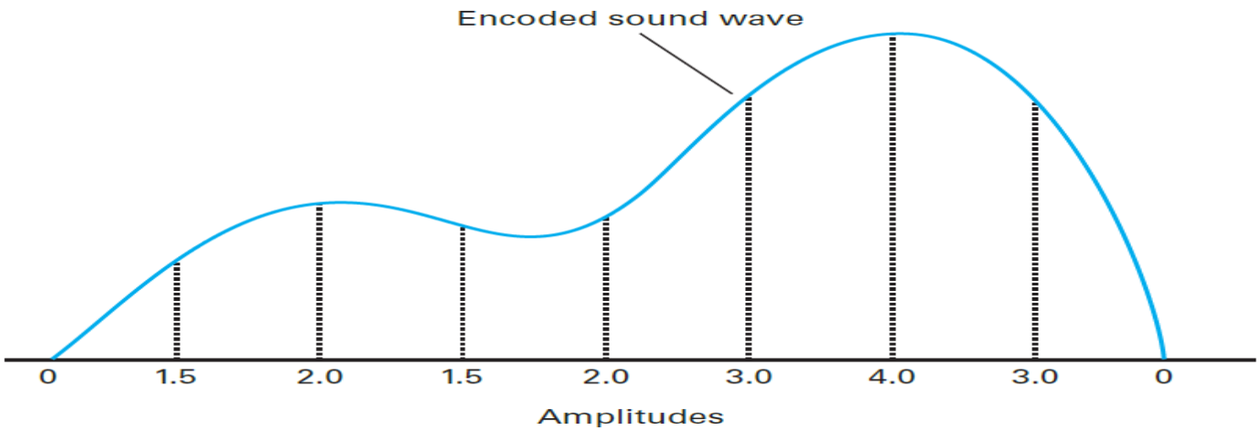
Sampling Amplitude

- ✓ Sample the amplitude of sound at regular intervals and records the values
- ✓ 8000 samples per second for long distance telephone communication

How communication happens

Sampling Amplitude

- ✓ At One end, it stores amplitude numeric values for each eight-thousand of a second
- ✓ Transmitted over the network
- ✓ Received on other end and sound is produced using amplitude.



Sample Intervals

Sampling Amplitude

- ✓ 8000 is not enough for high fidelity music recordings
- ✓ 44, 100 samples per second are recorded in todays CDs.
- ✓ Data from each sample is recorded in 16 bits (32 bits for Stereo)
- ✓ 32*44100 = million bits/sec

Alternative Method

MIDI

- ✓ Musical Instrument Digital Interface
- ✓ Used in music synthesizers found in electronic keyboards
- ✓ Encode directions for producing music rather than storing music itself.
- ✓ 2 seconds sound can be stored in 3 bytes rather than 2 million bits
- ✓ Encoding the sheet music read by performer rather than the performance itself
- ✓ MIDI recordings could be significantly different when performed on different synthesizer.

Summary

Representing Sound

- ✓ Sampling Amplitude
- ✓ MIDI

Topic-022. Binary notation

Recall Base 10

- ✓ 375
- ✓ Unit, tens, hundred and 10 to the power

Binary Numbers

- ✓ quantity associated with each position is twice the quantity associated with the position to its right

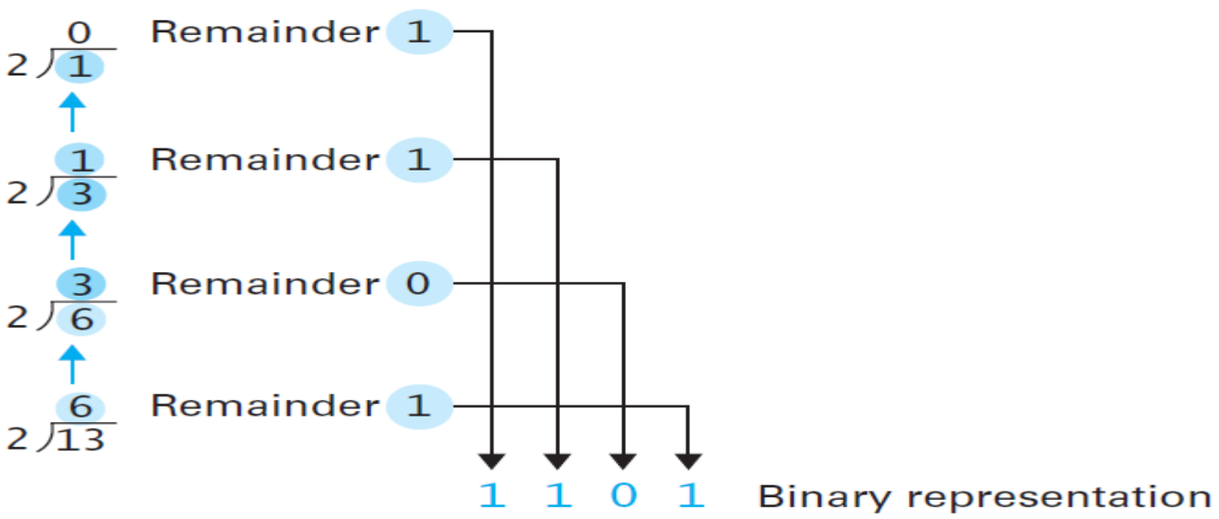
Positive Decimal Number

Step 1. Divide the value by two and record the remainder.

Step 2. As long as the quotient obtained is not zero, continue to divide the newest quotient by two and record the remainder.

Step 3. Now that a quotient of zero has been obtained, the binary representation of the original value consists of the remainders listed from right to left in the order they were recorded.

Applying Algorithm



Power Method Revision

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	Representing
128	64	32	16	8	4	2	1	
1	0	0	0	0	0	1	1	131
0	1	0	1	1	0	0	0	88
0	0	0	0	1	1	1	1	15

Summary

Binary Notation

- ✓ Division Method
- ✓ Power Method

Topic-023. Binary addition

Addition

Decimal

- ✓ 25 + 38
  Binary Addition

$$\begin{array}{r} 0 \\ +0 \\ \hline 0 \end{array} \quad \begin{array}{r} 1 \\ +0 \\ \hline 1 \end{array} \quad \begin{array}{r} 0 \\ +1 \\ \hline 1 \end{array} \quad \begin{array}{r} 1 \\ +1 \\ \hline 10 \end{array}$$

Examples

$$\begin{array}{r} 0 \\ +0 \\ \hline 0 \end{array} \quad \begin{array}{r} 1 \\ +0 \\ \hline 1 \end{array} \quad \begin{array}{r} 0 \\ +1 \\ \hline 1 \end{array} \quad \begin{array}{r} 1 \\ +1 \\ \hline 10 \end{array}$$

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	
128	64	32	16	8	4	2	1	
0	0	0	1	0	0	0	1	17
0	0	0	0	0	0	1	1	03
0	0	0	1	0	1	0	0	20

Dr. M. Tanvir Afzal

Examples

$$\begin{array}{r} 0 \\ +0 \\ \hline 0 \end{array} \quad \begin{array}{r} 1 \\ +0 \\ \hline 1 \end{array} \quad \begin{array}{r} 0 \\ +1 \\ \hline 1 \end{array} \quad \begin{array}{r} 1 \\ +1 \\ \hline 10 \end{array}$$

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	
128	64	32	16	8	4	2	1	
0	0	0	0	1	1	1	1	15
0	0	0	0	1	1	1	1	15
0	0	0	1	1	1	1	0	30

Dr. M. Tanvir Afzal

Examples

$$\begin{array}{r} 0 \\ +0 \\ \hline 0 \end{array} \quad \begin{array}{r} 1 \\ +0 \\ \hline 1 \end{array} \quad \begin{array}{r} 0 \\ +1 \\ \hline 1 \end{array} \quad \begin{array}{r} 1 \\ +1 \\ \hline 10 \end{array}$$

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	
128	64	32	16	8	4	2	1	
0	1	0	0	0	0	0	0	64
0	1	0	0	0	0	0	0	64
1	0	0	0	0	0	0	0	128

Dr. M. Tanvir Afzal

Examples

$$\begin{array}{r} 0 \\ +0 \\ \hline 0 \end{array} \quad \begin{array}{r} 1 \\ +0 \\ \hline 1 \end{array} \quad \begin{array}{r} 0 \\ +1 \\ \hline 1 \end{array} \quad \begin{array}{r} 1 \\ +1 \\ \hline 10 \end{array}$$

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	
128	64	32	16	8	4	2	1	
0	0	0	0	1	0	1	1	11
0	0	0	0	1	1	1	0	14
0	0	0	1	1	0	0	1	25

Dr. M. Tanvir Afzal

Examples

$$\begin{array}{r} 0 \\ +0 \\ \hline 0 \end{array} \quad \begin{array}{r} 1 \\ +0 \\ \hline 1 \end{array} \quad \begin{array}{r} 0 \\ +1 \\ \hline 1 \end{array} \quad \begin{array}{r} 1 \\ +1 \\ \hline 10 \end{array}$$

Page 56 of book

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	
128	64	32	16	8	4	2	1	
0	0	1	1	1	0	1	0	58
0	0	0	1	1	0	1	1	27
0	1	0	1	0	1	0	1	85

Dr. M. Tanvir Afzal

Summary

Binary Addition

- ✓ Rules
- ✓ Examples

Topic-024. Fraction in binary

Radix Point

- ✓ As we have decimal point
- ✓ Digits on left side represent the whole number
- ✓ Digits on right side represent the fractional part

Example

2^2	2^1	2^0	radix	2^{-1}	2^{-2}	2^{-3}
4	2	1	.	1/2	1/4	1/8
1	0	1		1	0	1

2^2	2^1	2^0	radix	2^{-1}	2^{-2}	2^{-3}
4	2	1	.	1/2	1/4	1/8
1	1	1		1	1	1

2^2	2^1	2^0	radix	2^{-1}	2^{-2}	2^{-3}
4	2	1	.	1/2	1/4	1/8
0	1	1		0	1	1

Addition in Fractional Numbers

1101.11
101.110

Align Radix

8	4	2	1	Radix	1/2	1/4	1/8	1/16
1	1	0	1	.	1	1	0	0
0	1	0	1	.	1	1	0	0

Dr. M. Tanvir Afzal

Addition in Fractional Numbers

1.0010
1000.01

Align Radix

8	4	2	1	Radix	1/2	1/4	1/8	1/16
0	0	0	1	.	0	0	1	0
1	0	0	0	.	0	1	0	0

Dr. M. Tanvir Afzal

Addition in Fractional Numbers

100.10
100.0001

Align Radix

8	4	2	1	Radix	1/2	1/4	1/8	1/16
0	1	0	0	.	1	0	0	0
0	1	0	0	.	0	0	0	1

Dr. M. Tanvir Afzal

Summary

Binary Fraction

- ✓ Representation
- ✓ Addition

Topic-025. 2's complement notation to store numbers

Integer Representations

- ✓ fixed number of bits to represent each value
- ✓ Normally 32 bits are used
- ✓ We will discuss smaller examples for demonstration purpose

Integer Representations

- ✓ For +ve integers, Starting from zero going upward until a single zero is reached followed by all 1s
- ✓ For -ve integers, Starting from all1s going downwards until a single 1 is reached followed by all 0s.
- ✓ Left most bit = sign bit.

a. Using patterns of length three

Bit pattern	Value represented
011	3
010	2
001	1
000	0
111	−1
110	−2
101	−3
100	−4

Integer Representations

b. Using patterns of length four

Bit pattern	Value represented
0111	7
0110	6
0101	5
0100	4
0011	3
0010	2
0001	1
0000	0
1111	−1
1110	−2
1101	−3
1100	−4
1011	−5
1010	−6
1001	−7
1000	−8

Conversion between +ve and –ve representations

- ✓ Start from right most bit start copying until 1st 1 arrives, after that complement all numbers.

b. Using patterns of length four

Bit pattern	Value represented
0111	7
0110	6
0101	5
0100	4
0011	3
0010	2
0001	1
0000	0
1111	−1
1110	−2
1101	−3
1100	−4
1011	−5
1010	−6
1001	−7
1000	−8

- ✓ All rules are same except that all bit patterns, including the answer, are the same length
- ✓ Truncation is performed

Problem in base 10	Problem in two's complement	Answer in base 10
3 + 2 →	0011 + 0010 = 0101 →	5
−3 + −2 →	1101 + 1110 = 1011 →	−5
7 + −5 →	0111 + 1011 = 0010 →	2

Problem of Overflow

- ✓ Using 4 bits, maximum +ve number 7 can be represented and -8 on the –ve side.
- ✓ 5+4 = 9 which can not be stored in four bits

- ✓ Result would be as -7.
- ✓ Can be deducted using sign bit

- ✓ Todays computer have longer bit pattern normally 32 bits
- ✓ Maximum positive value = 2,147,483,647
- ✓ If still overflow occurs, we can use even more bits, or changing the units like calculating answer in Km than meter.

Dangers of using Computer

- ✓ Previously 16 bits were used to store 2's complement notation, maximum number represented was 32,768
- ✓ On September 19, 1989, Hospital system malfunctioned as 32, 768 days had been passed after January 1 1900, thus it produced negative value.

Summary

2's Complement

- ✓ Representation of +ve and –ve numbers
- ✓ Conversions
- ✓ Addition in 2's complement
- ✓ Truncation error
- ✓ Overflow

b. Using patterns of length four

Bit pattern	Value represented
0111	7
0110	6
0101	5
0100	4
0011	3
0010	2
0001	1
0000	0
1111	−1
1110	−2
1101	−3
1100	−4
1011	−5
1010	−6
1001	−7
1000	−8

Topic-026. Excess Notation

Integer Representations

- ✓ fixed number of bits to represent each value
- ✓ Write down all bit patterns of the same length
- ✓ First bit pattern having 1 in the most significant bit is used to represent Zero
- ✓ Following values used to represent positive numbers and Preceding values to represent negative numbers
- ✓ Each value in Excess notation is Excess of its original value in Binary
- ✓ 1011 represent 11 in Binary but here it is representing 3 (excess of 8).
- ✓ Excess 16 to represent 10000 as Zero
- ✓ Excess 4 to represent 100 as Zero

- ✓ Excess 4 to represent

100 as Zero

- ✓ Values are Excess of 4.

Bit pattern	Value represented
1111	7
1110	6
1101	5
1100	4
1011	3
1010	2
1001	1
1000	0
0111	-1
0110	-2
0101	-3
0100	-4
0011	-5
0010	-6
0001	-7
0000	-8

Excess 4 comparison

Bit pattern	Value represented
111	3
110	2
101	1
100	0
011	−1
010	−2
001	−3
000	−4

Binary	Decimal	Excess 4	2's Complement
111	7	3	-1
110	6	2	-2
101	5	1	-3
100	4	0	-4
011	3	-1	3
010	2	-2	2
001	1	-3	1
000	0	-4	0

Summary

Excess Notation

- ✓ Representation of +ve and –ve numbers
- ✓ Easy to remember
- ✓ Conversions

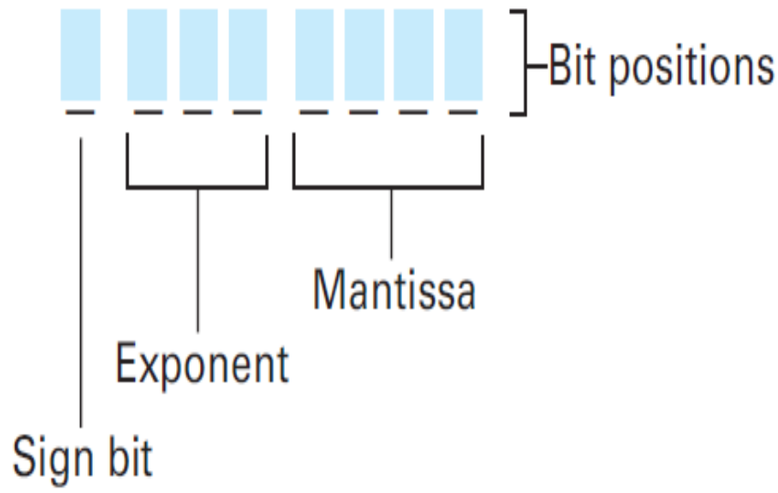
Topic-027. Floating Point Notation

Storing Radix

- ✓ Numbers with fractional part have a radix, so its important to store the position of Radix
- ✓ One popular way is floating-point notation
- ✓ For demonstration purpose we will use examples of 8 bits storage system.

Storing Fractions

- ✓ Sign bit
- ✓ Exponent field
- ✓ Mantissa field



Understanding Stored Fractions

- ✓ Suppose a number is stored 01101011

Understanding Stored Fractions

- ✓ Suppose a number is stored 00111100

Example to store data

Storing Fractions

- ✓ Suppose a number is stored 1 1/8

Storing Fractions

Normalized Form

- ✓ Suppose you want to store 3/8, the mantissa would be .011, but mantissa would be 1100 not 0110, so we start storing from first left 1.
- ✓ both 00111100 and 01000110 would be decoded as 3/8, the first one is in normalized form

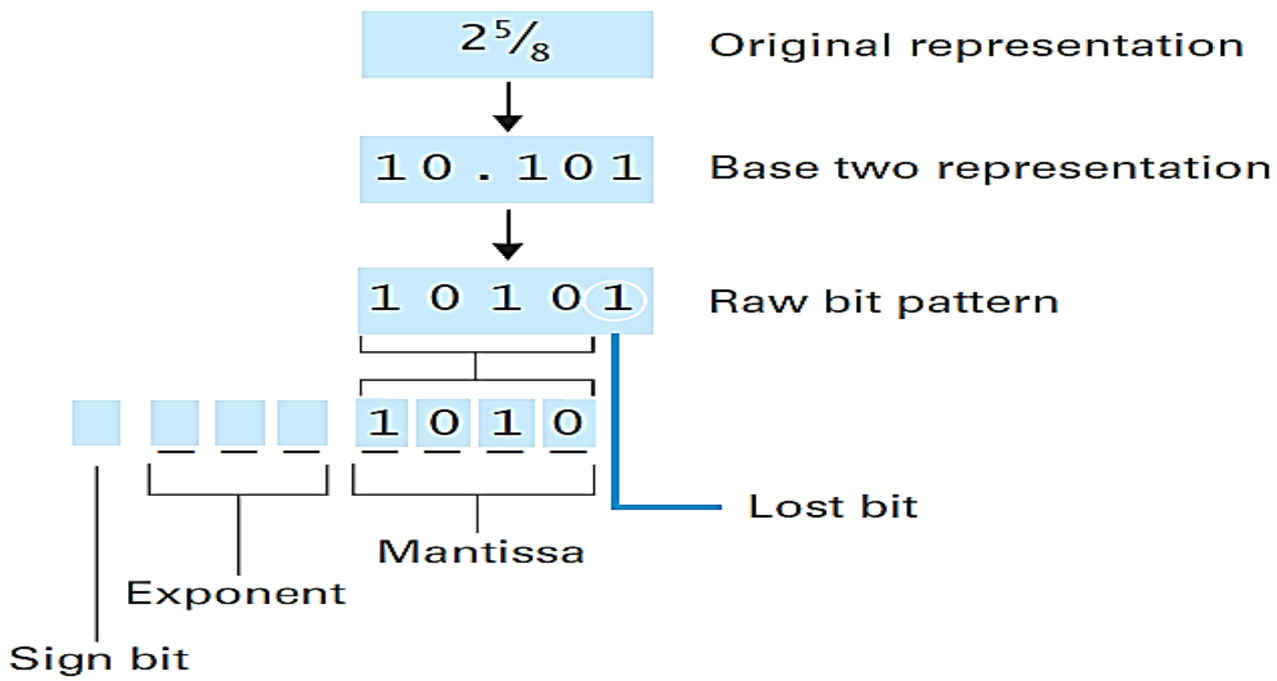
Summary

Floating Point Notation

- ✓ Storing fractions
- ✓ Sign bit, Exponent, Mantissa
- ✓ How to store and retrieve data in floating point

Topic-028. Truncation Errors in Floating Point Notation

Truncation Errors



Handling Truncation Errors

Primitive Methods

- ✓ Use more bits – todays computer use 32 bits

- ✓ Change the units
- ✓ Mathematics – Numerical Analysis

Intelligent Processing

- ✓ Lets suppose we want to store
- ✓ 2-½ + 1/8 + 1/8
- ✓ If we add 2-½ to 1/8 we ends up with 2-5/8 which is 10.101 which can not be stored in 4 bit mantissa.
- ✓ The result 10.10 would be 2-½ this means the 1/8 is truncated
- ✓ Lets add first 1/8 to 1/8 which is ¼ or .01, can be stored and result would be 00111000
- ✓ Now add this to 2-½ now we got 2-¾ = 01101011 which is accurate.
- ✓ Order is important, adding small quantities together first might give significant quantity to be added to large quantity.

Summary

Truncation errors in Floating points

- ✓ Why truncation errors occurs with examples.
- ✓ Primitive ways to deal
- ✓ Intelligent processing – the order of addition

Topic-029. Generic techniques

Generic Techniques

- ✓ Lossless (no loss of data)
- ✓ Lossy (may loss data)

Lossless Techniques

- ✓ Original data can be reconstructed from compressed data.
- ✓ Examples:
  - ✓ ZIP file format
  - ✓ Source Code
  - ✓ Text documents
  - ✓ Executable programs

Lossy Techniques

- ✓ Original data cannot be reconstructed from compressed data, only approximation is possible,
- ✓ Examples:
  - ✓ Multimedia data (audio, video, images)
  - ✓ Streaming media

Run Length Encoding

- ✓ Suitable where the long sequence have same value like 900 ones followed by 300 zeros

Frequency Dependent encoding

- ✓ Length of the bit pattern used to represent a data item is inversely related to the frequency of the item's use
- ✓ Variable Length codes
- ✓ Huffman Codes.

Huffman Codes

- ✓ In English language characters 'e, t, a, i' are more frequent than 'z, q, x', So using smaller bit patters to represent: 'e, t, a, i' would save space.

Relative Encoding/Differential Coding

- ✓ Stream of data differs only slightly from the previous ones. Like motion picture.
- ✓ Record differences rather than storing the whole frame again

Dictionary Encoding

- ✓ Store just a reference to a dictionary term not the whole word.
- ✓ Especially used by word processors as they have already dictionary for spell check.

Adaptive Dictionary Encoding

- ✓ Dictionary dynamically changes over time, when a larger unit is found, it is added in the actual dictionary.
- ✓ One example is: Lempel-Ziv-Welsh (LZW) encoding
  Applying LZW technique

Example

xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz
Summary

Compression Techniques

- ✓ Generic data compression techniques
- ✓ Lossless and Lossy techniques
- ✓ LZW with example

Topic-030. Compressing images

Images

- ✓ Bitmap were studied in previous modules
- ✓ They can further be optimized to store images

GIF

- ✓ Graphic Interchange Format
- ✓ Pronounced as “Giff” of “Jiff”
- ✓ Developed by CompuServe
- ✓ It reduces the number of colors to 256 that can be assigned to all combinations of RGB.
- ✓ 3 bytes to 1 byte.
- ✓ Lossy technique.
- ✓ GIF further applies LZE (Adaptive dictionary system)

GIF: Pros and cons

- ✓ One of the color in GIF is transparent which makes it a choice for simple animations
- ✓ However, it is unreliable for higher precision systems like photography.

JPEG

- ✓ Standard developed by Joint Photographic Expert Group.
- ✓ Widely used for Photography industry as well
- ✓ Many options of lossless and lossy approaches
- ✓ JPEG lossless does not achieve much compression
- ✓ JPEG baseline standard (lossy method) is normally used.

JPEG baseline

Step 1: Take advantage of human eye’s limitation.

- ✓ We are more sensitive to see changes in brightness than the change in color.
- ✓ It exploits chrominance values.

Step 2: divide the image in 8 * 8 pixel block.

- ✓ Discrete cosine transformation – forms another block that tell how the values are stored rather than the actual values.
- ✓ At least 10% reduction or upto 30% reduction

Summary

Compressing Images

- ✓ Bitmap to GIF
- ✓ GIF to JPEG

- ✓ Algorithm,
- ✓ Pros and cons

Topic-031. Compressing audio & videos

Formats

- ✓ MPEG
- ✓ MP3

MPEG

- ✓ Motion Picture Expert Group lead by ISO.
- ✓ Various standards for different purposes:
- ✓ HDTV vs video conferencing

MPEG Techniques

- ✓ Considering video as a sequence of pictures
- ✓ Some of the pictures known as I-frames are encoded entirely.
- ✓ Pictures between I-frames are encoded using relative encoding techniques (distinctions from the previous image)

MP3

- ✓ Developed within the MPEG.
- ✓ MP3 = MPEG layer 3.
- ✓ **MP3 working**
- ✓ Taking advantage from human ear.
- ✓ Temporal Masking: for a short period after a loud sound, we can not hear soft sound.
- ✓ **Frequency Masking:** sound at one frequency tend to mask softer sounds at nearby frequencies.

MPEG and MP3

- ✓ Video cameras are able now to record hours worth video in 128MB.
- ✓ 400 popular songs on a single GB.
- ✓ Goal is not just reducing the space.
- ✓ Making an encoding that allow data transmission faster.

Data Transmission: MPEG and MP3

- ✓ Normally measure in bits per second. Kbps (1000 bits), Mbps (1 million bits), Gbps (1 billion bits).
- ✓ MPEG – successfully relayed on communication paths having 40 Mbps transfer rate.
- ✓ MP3 do not require transfer rate of more than 64 Kbps

Summary

Compressing Audio/Videos

- ✓ MPEG and MP3
- ✓ Space
- ✓ Transfer rate

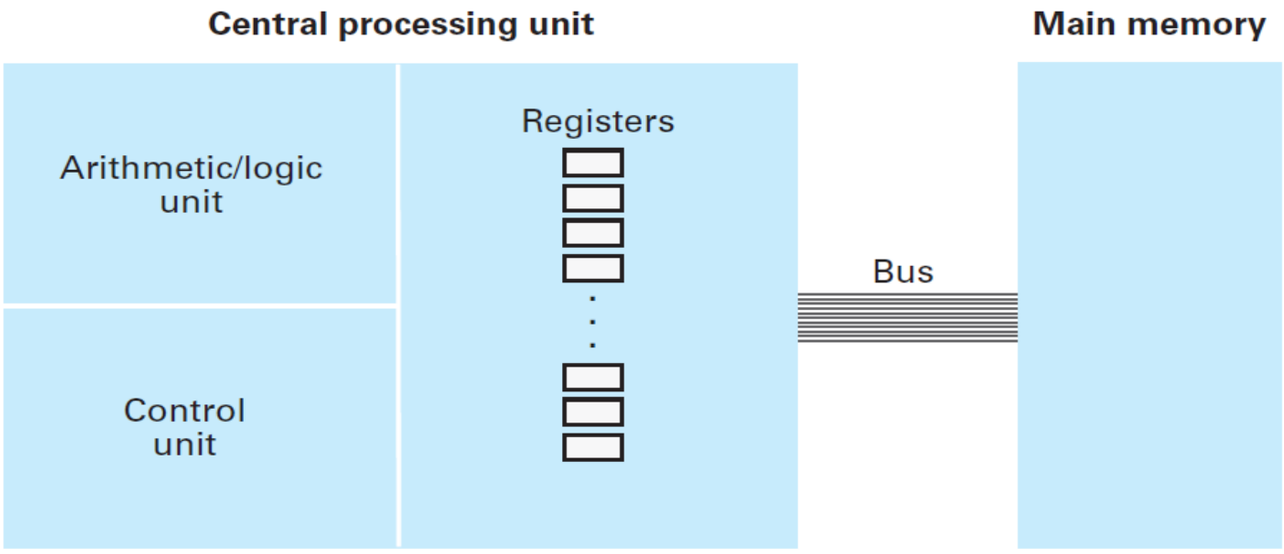
Topic-032. CPU basics

What is CPU?

- ✓ The circuitry in a computer that controls the manipulation of data is called the central processing unit, or **CPU (Processor)**

Size of CPU

- ✓ In the mid-twentieth, it comprises of large racks
- ✓ Today 2*3 inch connected to motherboard.
- ✓ In smartphones, the size is as small as half of the postal stamp.



CPU Components

- ✓ **ALU:** contains circuitry that performs operations like addition and subtraction
- ✓ **CU:** contains circuitry for coordination of machine activity.
- ✓ **Register Unit:** temporary storage of information.

Adding values stored in memory

Step 1. Get one of the values to be added from memory and place it in a register.

Step 2. Get the other value to be added from memory and place it in another register.

Step 3. Activate the addition circuitry with the registers used in Steps 1 and 2 as inputs and another register designated to hold the result.

Step 4. Store the result in memory.

Step 5. Stop.

Summary

CPU Basic

- ✓ ALU
- ✓ CU
- ✓ Registers
- ✓ Main memory
- ✓ Addition Algorithm

Topic-033. Stored program

Early computers

- ✓ Not flexible
- ✓ Steps (Algorithm) was part of the CPU.
- ✓ Next thing was to rewire the CPU
- ✓ A breakthrough of storing the program in Main memory. incorrectly credited to John Von Neumann.

Recall Addition Algorithm

Adding values stored in memory

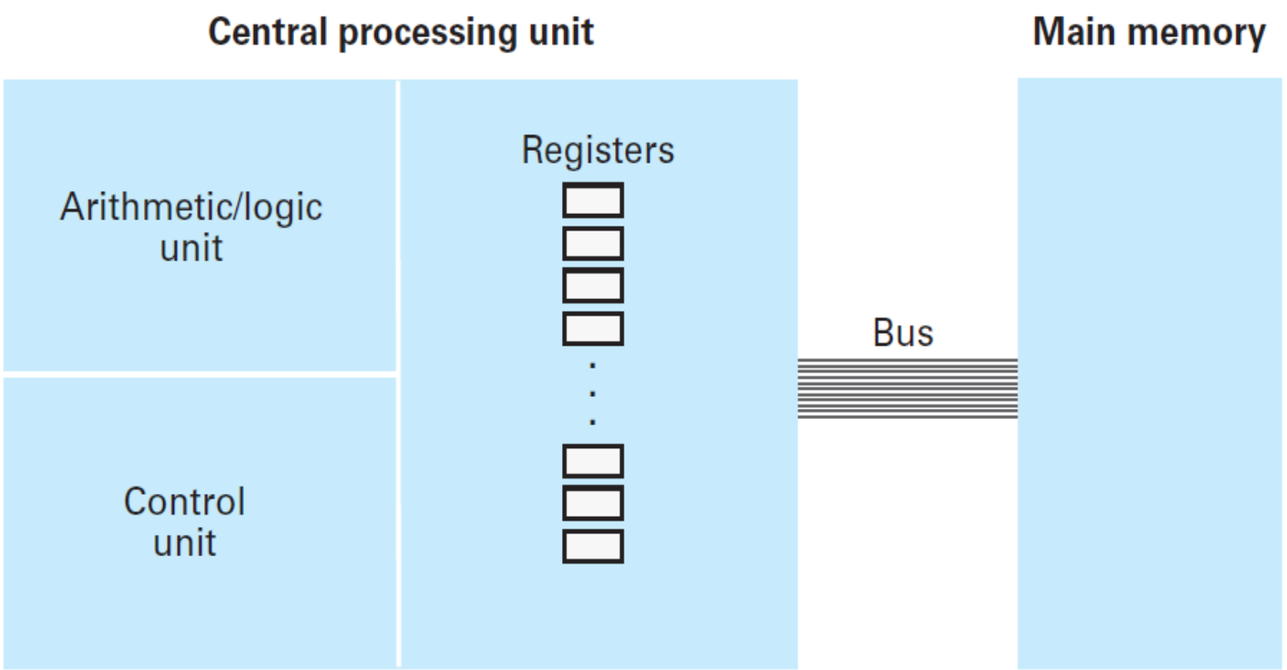
Step 1. Get one of the values to be added from memory and place it in a register.

Step 2. Get the other value to be added from memory and place it in another register.

Step 3. Activate the addition circuitry with the registers used in Steps 1 and 2 as inputs and another register designated to hold the result.

Step 4. Store the result in memory.

Step 5. Stop.



Summary

Stored Program

- ✓ Putting algo within CPU
- ✓ Rewiring of CPU
- ✓ Stored programs

Topic-034. CPU architecture philosophies

Machine Instruction

- ✓ For Stored program, CPUs are designed to recognize the instructions encoded as bit patters, Machine Language, Machine Instruction.

Basic Machine Instructions

- ✓ Machine Instructions are quite short, once basic level is met, adding more instructions is not adding anything in machine's capability, its just making it more convenient.

RISC

- ✓ Reduced Instruction Set Computer.
- ✓ Should execute minimal set of instructions.
- ✓ Machine is efficient, fast and less expensive.

CISC

- ✓ Complex Instruction set Computer
- ✓ Others argue CPU should be able to execute large complex instructions even redundant .
- ✓ The more complex CPU can cope with the ever increasing complexities

CISC Vs RISC

- ✓ In 1900 and even in millennium, both architectures were competing to take place in desktop computers
- ✓ Intel Processors developed for PC are CISC,
- ✓ Power PC by Apple, IBM, and Motorola are RISC.

CISC viability

- ✓ The cost to build CISC reduced and intel processors are being used virtually in PCs, laptops, even Apple is using intel at backend.

RISC utilization

- ✓ Company Advanced Risk Machines (ARM) utilized RISC for low power consumption in game controllers, digital TVs, navigation systems, smartphones

Summary

CPU Architecture Philosophies

- ✓ CISC
- ✓ RISC
- ✓ Their realization

Topic-035. Machine Instruction Categories

Machine Instruction

- ✓ Whatever architecture is used: CISC or RISC, machine instructions can be categorized into three broad classes.

Classes:

- ✓ Data Transfer group
- ✓ Arithmetic/Logic group
- ✓ Control group

Data Transfer Group:

- ✓ Deals with transfer of data.
- ✓ Step 1, 2, 4.
- ✓ Transfer is not moving, its copying rather.
- ✓ Special terms are used when talking about transfer between CPU and memory.

Adding values stored in memory

Step 1. Get one of the values to be added from memory and place it in a register.

Step 2. Get the other value to be added from memory and place it in another register.

Step 3. Activate the addition circuitry with the registers used in Steps 1 and 2 as inputs and another register designated to hold the result.

Step 4. Store the result in memory.

Step 5. Stop.

Data Transfer Group:

- ✓ LOAD: retrieving data from memory and filling the general purpose register.
- ✓ STORE: Register to memory
- ✓ I/O instructions: instructions for external devices (Printer, Scanner, Keyboard)

Arithmetic/Logic Group:

- ✓ Request an activity within ALU.
- ✓ Step 3
- ✓ Boolean operations like AND, OR, XOR
- ✓ SHIFT. ROTATE,

Adding values stored in memory

Step 1. Get one of the values to be added from memory and place it in a register.

Step 2. Get the other value to be added from memory and place it in another register.

Step 3. Activate the addition circuitry with the registers used in Steps 1 and 2 as inputs and another register designated to hold the result.

Step 4. Store the result in memory.

Step 5. Stop.

Control Group:

- ✓ Execution rather than manipulation
- ✓ Step 5.
- ✓ Many other instructions like JUMP (BRANCH),
- ✓ Unconditioned Jump

- ✓ Conditional Jump.

Adding values stored in memory

Step 1. Get one of the values to be added from memory and place it in a register.

Step 2. Get the other value to be added from memory and place it in another register.

Step 3. Activate the addition circuitry with the registers used in Steps 1 and 2 as inputs and another register designated to hold the result.

Step 4. Store the result in memory.

Step 5. Stop.

Dividing values stored in memory

Step 1. LOAD a register with a value from memory.

Step 2. LOAD another register with another value from memory.

Step 3. If this second value is zero, JUMP to Step 6.

Step 4. Divide the contents of the first register by the second register and leave the result in a third register.

Step 5. STORE the contents of the third register in memory.

Step 6. STOP.

Summary

Machine Instructions categories

- ✓ Data Transfer group
- ✓ Arithmetic/Logic
- ✓ Control
- ✓ Examples and scenarios

Topic-036. Program execution

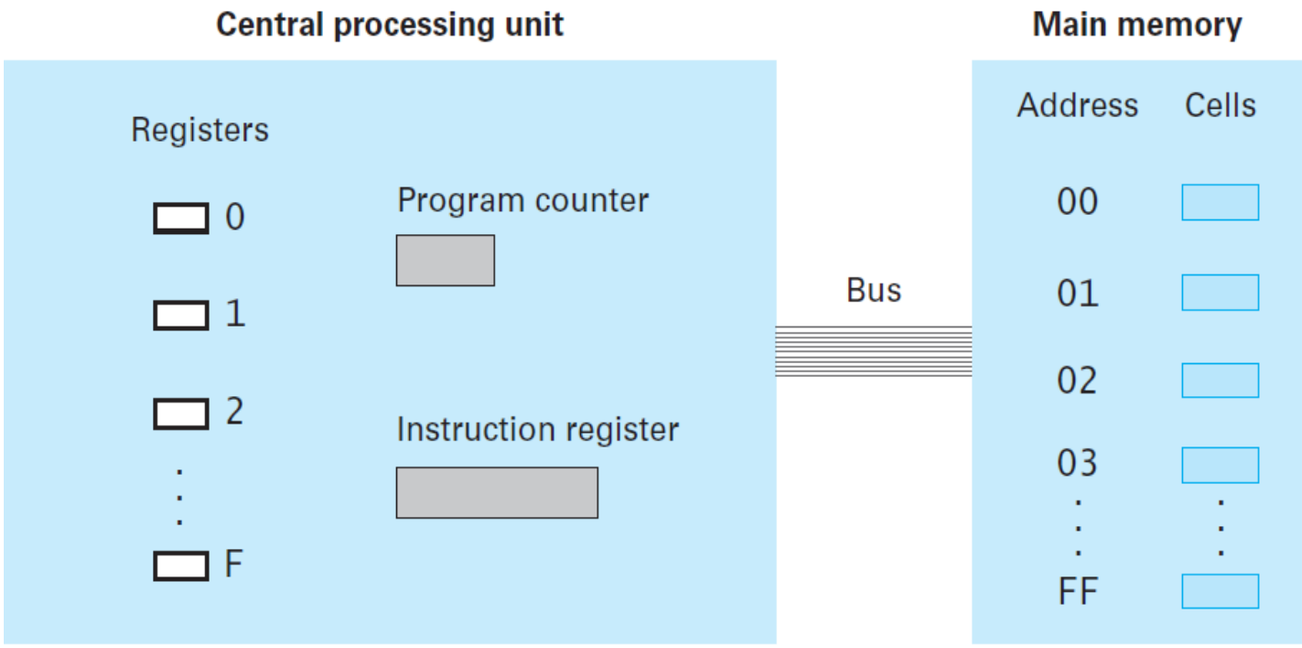
Machine Instruction

- ✓ From main memory to CPU
- ✓ Each instruction is decoded and obeyed.
- ✓ The order is as the instructions are stored in memory otherwise specified by a JUMP.

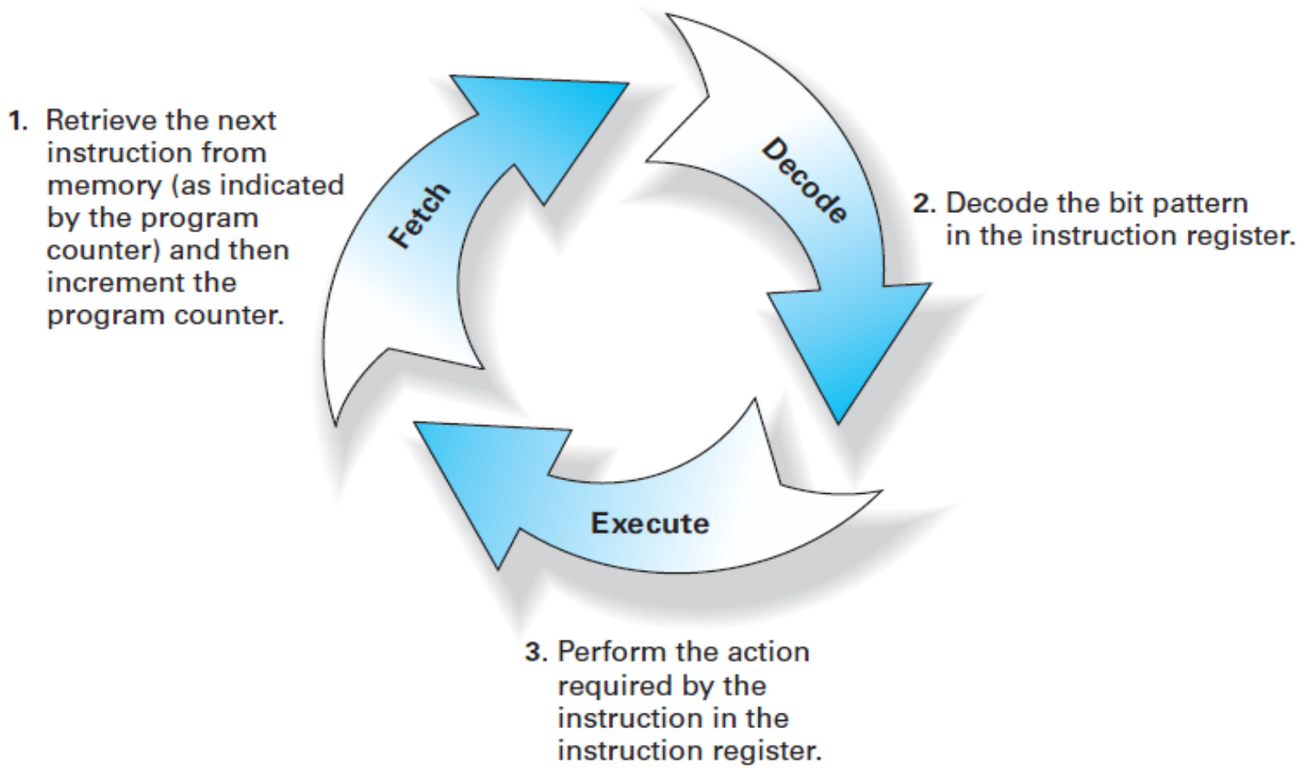
Special Purpose Registers

- ✓ **Instruction Register:** Holding instruction being executed now.

✓ **Program Counter:** contains the address of next instruction to be executed.



Machine Cycle



Summary

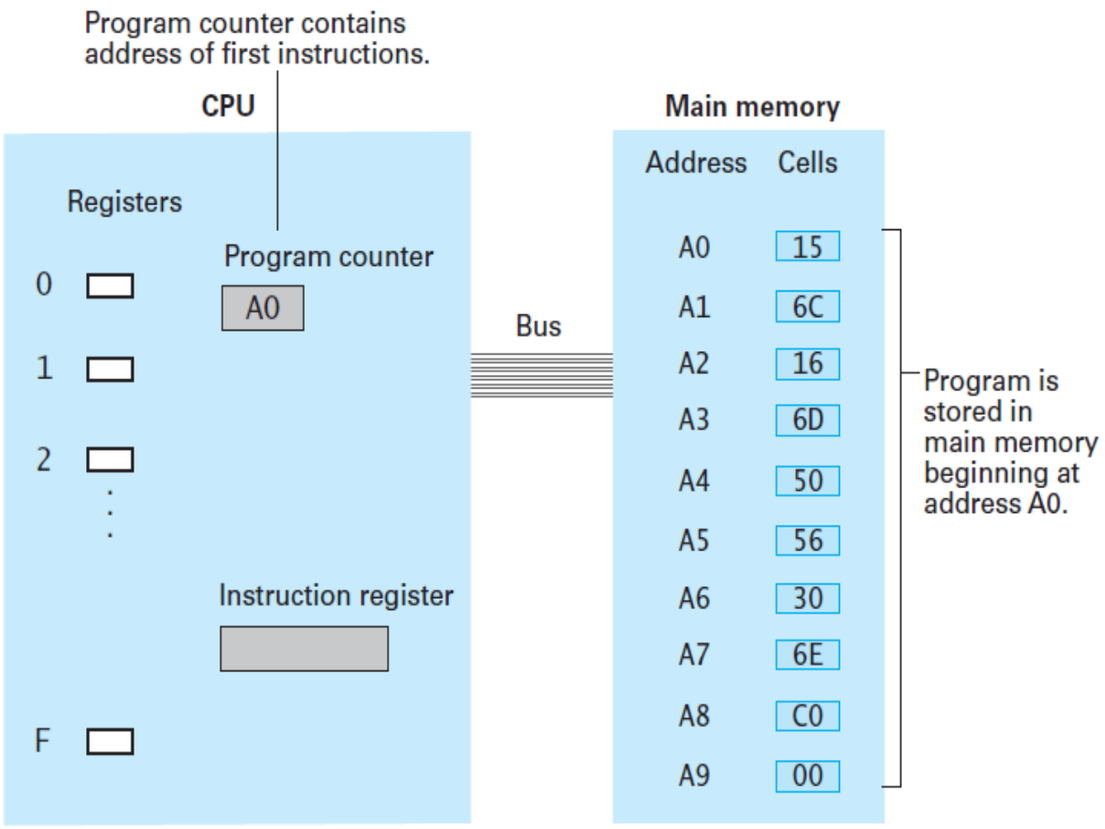
Program Execution

- ✓ Special purpose register
- ✓ Machine Cycle

Topic-037. Program execution example

Machine Instruction

- ✓ In your book an example of machine instructions with their meanings are available in Appendix C at page 581. lets discuss it
- ✓ Lets execute a program that reads two numbers from memory, adds them and store in the memory.



Summary

Program Execution Example

- ✓ Appendix C
- ✓ Solved an example by execute an addition algorithm using Machine instructions

Topic-038. Logic operators

ALU

- ✓ Arithmetic
- ✓ Logic
- ✓ Shift

Logic Operators

- ✓ Bitwise operators
- ✓ AND
- ✓ OR
- ✓ XOR

Why we use Logic Operators: AND

- ✓ Placing zeros in one part of the bit patterns while not changing the others
- ✓ When we want to eliminate Red from RGB
- ✓ 16 bits showing the availability of Ice cream flavor
- ✓ This is called Masking
- ✓ When we want to find out whether particular flavor of ice cream is available or not

Why we use Logic Operators: OR

- ✓ Putting 1s at designated bits in the bit pattern without changing the remaining.
- ✓ mask 11011111 can be used with AND to make sure the third bit becomes zero
- ✓ mask 00100000 can be used with OR to make sure the third bit becomes zero

Why we use Logic Operators: XOR

- ✓ Forming the complement of the bit pattern by XORing with all 1's.
- ✓ Can invert an image.

Summary

Logic Operators

- ✓ Bitwise operators
- ✓ AND
- ✓ OR
- ✓ XOR

- ✓ Practical scenarios

Topic-039. Rotation and shift

ALU

- ✓ Moving bits within the register (right or left)
- ✓ Example:

Circular Shift/Rotation

- ✓ Consider a byte 10010101
- ✓ One bit right, and rightmost should be put at the hole created on the left side.

Logical Shift

- ✓ Consider a byte 10010101
- ✓ One bit right, and rightmost should be discarded and on left side we put Zero.
- ✓ On left side for 2's complement notation multiplying the value
- ✓ As decimal values are multiplied by 10 when shifted.
- ✓ We should take care of sign bit – such shifts sometimes are called Arithmetic Shifts.

Summary

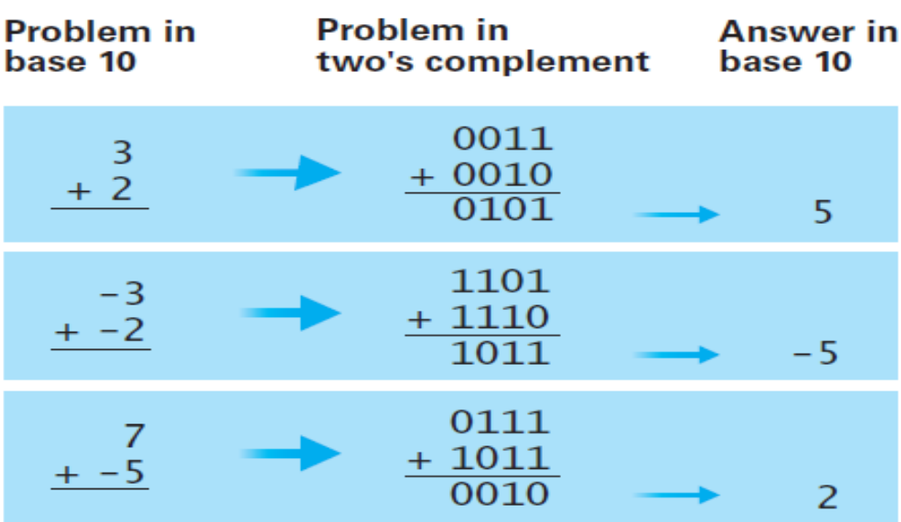
Rotation and Shifting

- ✓ Left/right rotation and shifting
- ✓ Multiplication and division
- ✓ Arithmetic Shifts.

Topic-040. Arithmetic operators

Addition and Subtraction

- ✓ Subtraction can be simulated by addition and negation like in 2's complement notation 7 -5 would be 7 + (-5) which means the binary of 7 will be added to binary of -5.



Multiplication

- ✓ Multiplication is repetitive addition
- ✓ For example 8 multiplied by 3, we will add 8 three times

Division

- ✓ Can be achieved through subtraction
- ✓ 15-5
- ✓ Some small CPUs are designed to have just add, or just add and subtract t do all arithmetic operations.

Remember addition

- ✓ If its stores as 2's complement, it is straight forward
- ✓ If its save in floating point notation, then you know you need to use mantissa, exponent and sign bit.
- ✓ Although both are additions but both have different workout.

Summary

Arithmetic Operations

- ✓ Addition and subtraction
- ✓ Multiplication is repetitive addition
- ✓ Division is repetitive subtraction
- ✓ Remember how the data was stored.

Topic-041. Role of controller

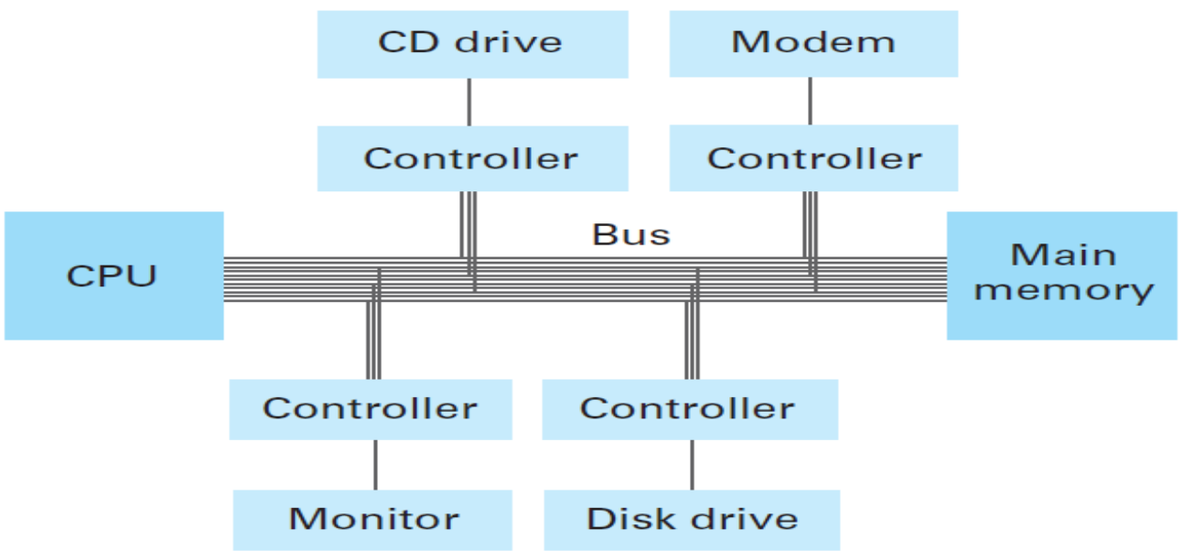
Communicating with other Devices

- ✓ Main memory and CPU form the core
- ✓ How they communicate with peripheral devices like printer, scanner, keyboard etc.
- ✓ Role of Controllers
- ✓ DMA
- ✓ Handshaking
- ✓ Communication media/Rates

What is Controller

- ✓ Communication between computer and other devices.
- ✓ In case of PC, it is permanently mounted on motherboard.
- ✓ Or flexible to be attached.
- ✓ It connects via cables to peripheral devices or to a connector called **Port**.
- ✓ They are themselves a small computer having their own CPU.
- ✓ Controller translates the messages back and forth.
- ✓ Originally each controller worked for only one device
- ✓ Now standard like USB (Universal Serial Bus).
- ✓ Firewire.

How it works



Summary

Role of Controller

- ✓ What is controller.
- ✓ How it works.

- ✓ Ports
- ✓ Dedicated Controller
- ✓ USB Firewire

Topic-042. Direct memory access and handshaking

Direct Memory Access

- ✓ Since Controller is attached to computer bus.
- ✓ It can communicate to main memory in those nanosecond when CPU is using bus.
- ✓ This ability of computer to access main memory is called **DMA**

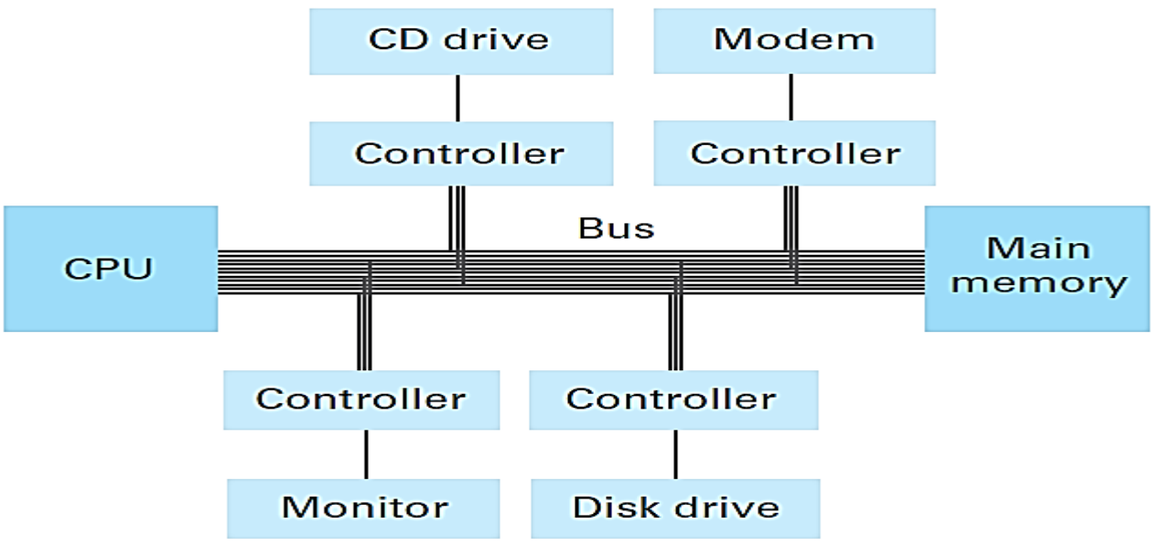
Why DMA

- ✓ It is time consuming to involve CPU for I/O and CPU is waiting for the I/O and supervising whether the I/O has been performed or not.

Scenario

- ✓ CPU asks controller to read data from some peripheral device or from one portion of the main memory and place it at a designated place in the memory.
- ✓ In the meanwhile CPU is free to perform other operations.
- ✓ Resource efficient.

Scenario in action



Von Neumann Bottleneck

- ✓ CPU and main memory
- ✓ CPU and each controller
- ✓ Each controller and main memory.
- ✓ CPU and controller are competing for the bus access.

Handshaking

Why we need

- ✓ CPU wants to send data to printer
- ✓ CPU speed is faster than printer could print them
- ✓ We need a constant handshake, acknowledgement where the peripheral device has reached.

How it works

- ✓ Peripheral device like printer can constantly telling its status to CPU like work done, paper jam, out of paper etc in the form of a bit pattern.

Summary

Direct Memory Access and Handshaking

- ✓ Controller accessing the main memory.
- ✓ Benefits for CPU
- ✓ Von Neumann Bottleneck
- ✓ Handshaking

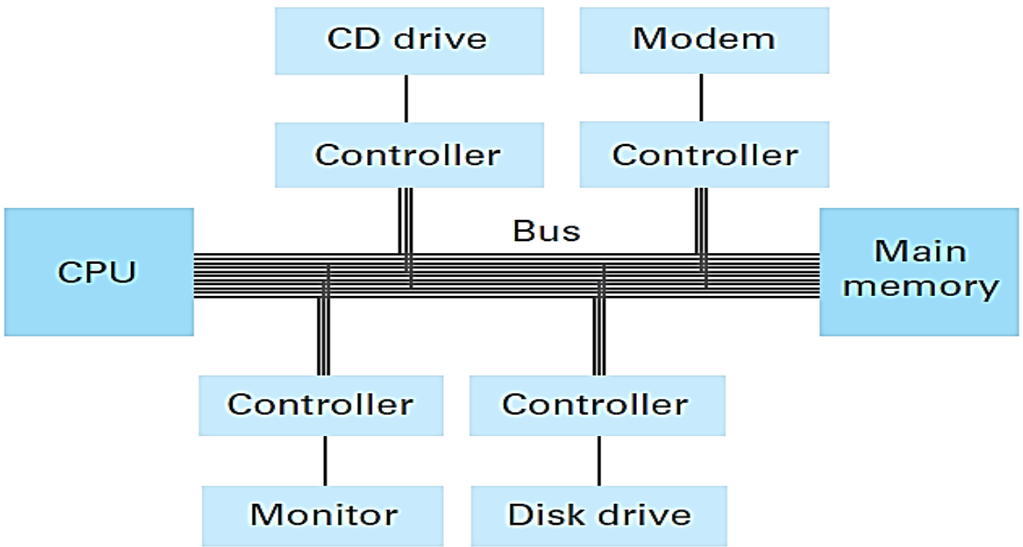
Topic-043. Communication media and communication rates

Communication Media

- ✓ Communication between computing devices is handled over two types of paths:
- ✓ **Parallel communication**
- ✓ **Serial Communication**

Parallel Communication

- ✓ Several signals are transferred at the same time, each on a separate line.
- ✓ Data transfer is good
- ✓ Requires complex architecture



Serial Communication

- ✓ Transfer data one after the other,
- ✓ Requires a simple data path
- ✓ Data transfer rate is relatively slower than parallel communication.
- ✓ USB and FireWire are examples of high-speed data transfer over a short distance.
- ✓ Ethernet connections for slightly longer distances.

Communication over longer distances

- ✓ Traditional voice lines dominated the PC arena for many years.
- ✓ Converting bit patterns to audible tones using Modem (Modulator-demodulator)
- ✓ For faster long distance communication, DSL (Digital Subscriber Line)

Cable Modems:

- ✓ Fibre Optics and Coaxial cables for high definition TV and computer network.

Communication Rates

- ✓ Rate at which bits are transferred from one computing component to another measured in bits per second **(bps)**
- ✓ **Kbps** (Kilo bits)
- ✓ **Mbps** (Million bits)
- ✓ **Gbps** (Billion bits)

- ✓ 8kbps = 1KB per second.
- ✓ USB and Firewire provide several hundred Mbps.

Summary

Communication media and communication rates

- ✓ Serial communication
- ✓ Parallel communication
- ✓ Communication rates

Topic-044. Pipelining

Electric pulses

- ✓ Electric pulses travel through wire no faster than the speed of light.
- ✓ Since light travels 1 foot in a nanosecond (one billionth second)
- ✓ CPU requires at least 2 nanosecond to fetch the instruction from memory.
- ✓ Read request to memory and instruction sent back.

Throughput

- ✓ Fetching, decoding and executing requires several nano-second.
- ✓ Increasing execution speed is not the only way to increase the CPU throughput (The amount of work done by a machine in a given time)

Increasing Throughput

- ✓ Increasing throughput without increasing the speed is achieved through Pipelining.
- ✓ Allow steps in machine cycle to overlap.

Pipelining vs without Pipelining

Pipelining			
Time	Fetch	Decode	Execute
T1	1		
T2	2	1	
T3	3	2	1
T4	4	3	2
T5	5	4	3
T6		5	4
T7			5

Without PipeLinie			
Time	Fetch	Decode	Execute
T1	1		
T2		1	
T3			1
T4	2		
T5		2	
T6			2
T7	3		
T8		3	
T9			3
T10	4		
T11		4	
T12			4
T13	5		
T14		5	
T15			5

Modern Machines

- ✓ Modern Machines are even able to fetch several instructions at one time.
- ✓ Even execution of instructions at one given time when the instructions are not rely on each other.

Summary

Pipelining

- ✓ Increasing Throughput of Computer
- ✓ Pipelining concept
- ✓ Examples
- ✓ Modern computers characteristics

Topic-045. History of OS

What is OS

- ✓ An operating system is the software that controls the overall operation of a computer.
- ✓ Store and retrieve files
- ✓ Execution of programs

- ✓ Program scheduling
- ✓ Memory management

Examples of OS

- ✓ Windows
- ✓ UNIX
  - ✓ Mac OS
  - ✓ Solaris

Early OS

- ✓ In 1940's and 1950's computer occupied the whole room.
- ✓ Program execution required significant preparations like: mounting magnetic tapes, placing punch cards in card readers, setting switches

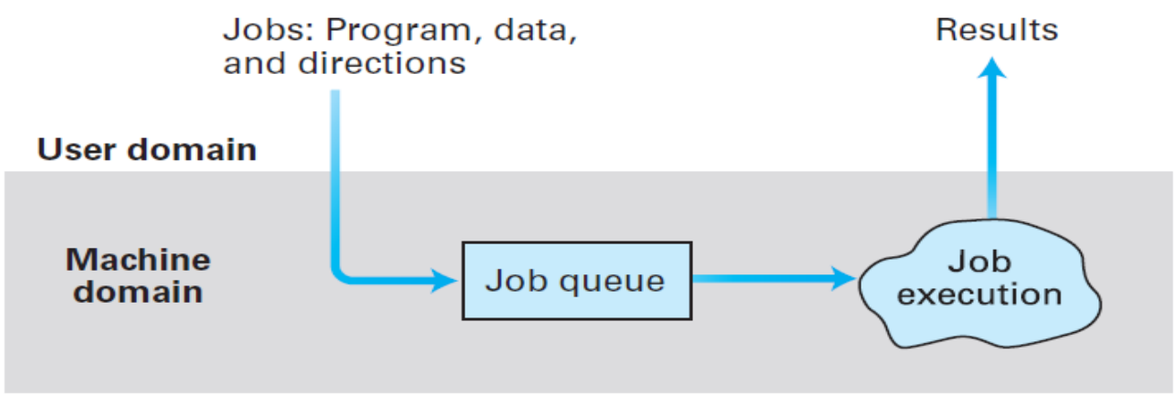
Job

- ✓ Execution of each program called the Job was isolated activity.
- ✓ a machine was shared between multiple users in different time slots.
- ✓ Then OS began as systems for simplifying program setup and streamlining the transition between jobs.

Beginning of OS

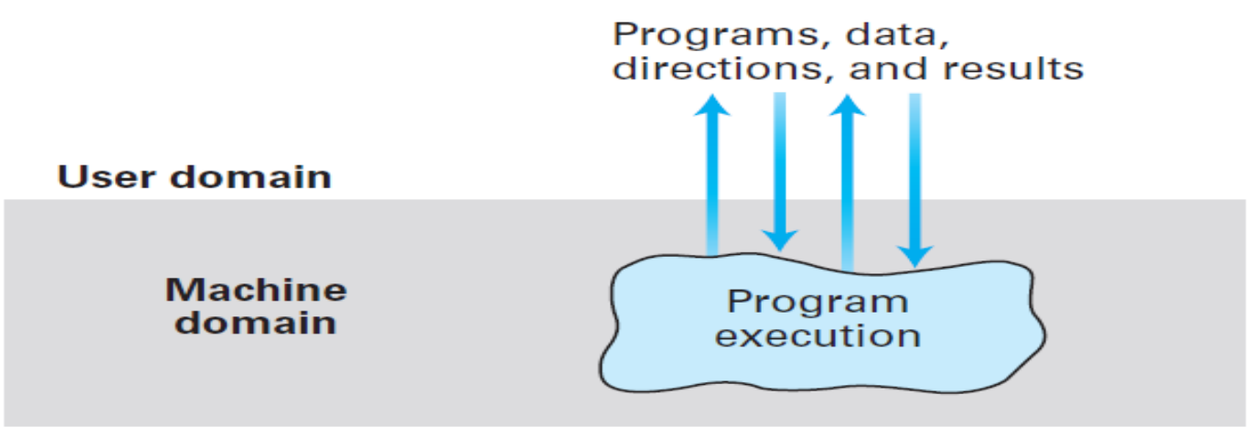
- ✓ Separation of users and transition.
- ✓ Computer Operator was hired to operate the machine.
- ✓ User submit the data and directions for the program to operator who load the material in mass storage
- ✓ OS was able to read one job after another : **Batch Processing**

Batch Processing



- ✓ Each job reside in a queue to be executed on its turn – **First In First Out.**
- ✓ Drawback: Users were not able to interact with their programs.

Interactive Processing



- ✓ Dialogue with the user through remote terminals.
- ✓ Terminals have just more then the type-writer

Summary

History of OS

- ✓ Early OS
- ✓ Batch Processing
- ✓ Interactive Processing

Topic-046. OS basic concepts-1

Coordination with User

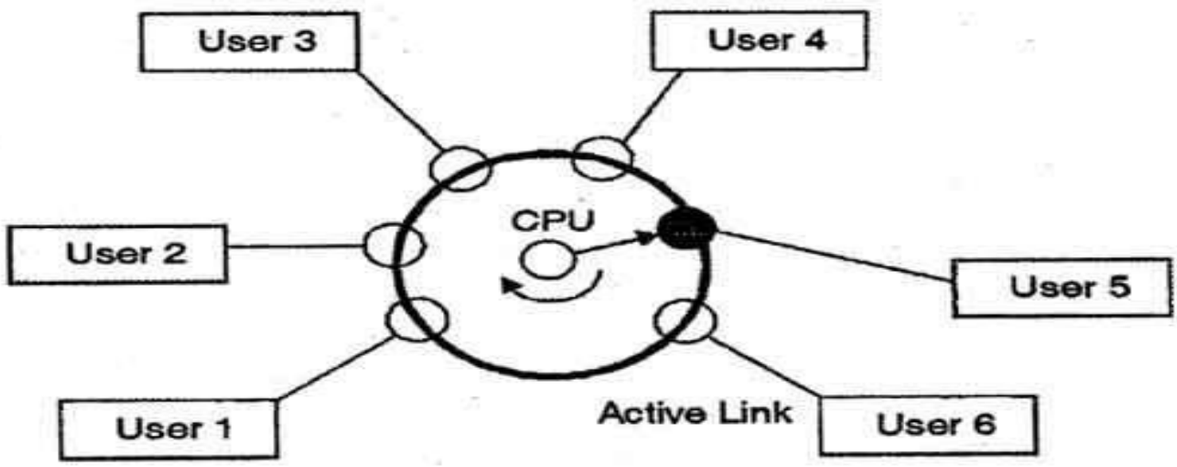
- ✓ Successfully Interactive processing is to respond the users with sufficiently fast time.
- ✓ Printing record of all students VU versus word processing (typing characters)
- ✓ Execution of tasks under a deadline

Real-time Processing

- ✓ Computer performs the tasks in accordance with the deadlines in its external real-world environment.
- ✓ Example of Cruise Missile, Radar etc

Interactive system and Real-time Processing

- ✓ If a system is servicing only one user, real-time processing was relatively easier to implement, however, computers in 60s, 70s were expensive,
- ✓ Each machine had to serve more than one user at remote terminals.



- ✓ Based on this problem, such OS were designed to service multiple users at the same time called time-sharing. Active, Ready, Waiting states

Multi-programming

- ✓ One way of implementing time-sharing, small time intervals, each job is executed for such a small interval.
- ✓ Early systems were able to service 30 users simultaneously.

Multitasking

- ✓ One user executing several tasks simultaneously.
- ✓ Today Multiprogramming is used in single user and multiple users

Summary

OS Basic Concepts (I)

- ✓ Realtime processing
- ✓ Time Sharing
- ✓ Multi-programming
- ✓ Multi-tasking

Topic-047. OS basic concepts-2

Computer Installations

- ✓ After Multiuser, time-sharing OS, central computer with remote workstations
- ✓ User could communicate directly to computer rather than computer operator.

- ✓ Commonly used programs resides in the workstation mass storage.

System Administrator

- ✓ Operator was no more needed to communicate between user and computer.
- ✓ Today's PC are on sole discretion of user
- ✓ Obtaining and Overseeing installations
- ✓ Issuance of new accounts
- ✓ Establishing Mass storage space limit
- ✓ - Rather than operating the machines in a hands-on manner.

OS Evolution

- ✓ From simple programs that execute one job at a time to complex systems having time-sharing, multiprogramming, multitasking, memory management.
- ✓ This evolution further continued by providing:
- ✓ Time sharing, multitasking by assigning different tasks to different processors, or different tasks to one processor.
- ✓ **Load Balancing**: Dynamically allocating tasks to different processors.

Scaling: breaking tasks into a number of subtasks compatible with the number of processors available.

Computer Networks are then helpful of managing resources across many machines and many users rather than an isolated computer.

Embedded Systems

- ✓ Medical devices, vehicle electronics, home appliances, cell phones, or other hand-held computers.
- ✓ Conserve battery power and working continuously in real-time fashion.

Summary

OS Basic Concepts (II)

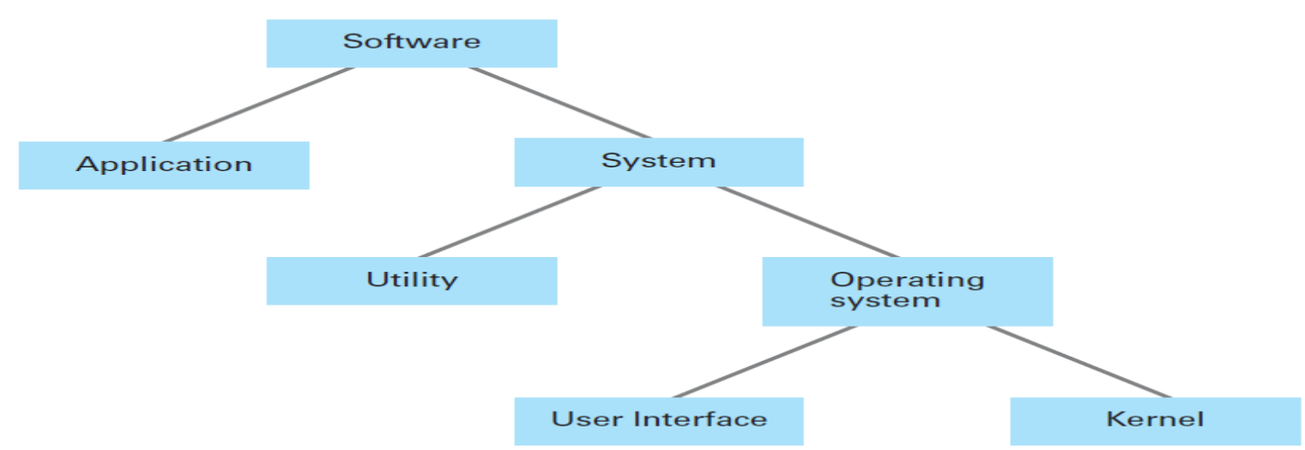
- ✓ The rule of human expert
- ✓ System administrator
- ✓ Load Balancing
- ✓ Scaling
- ✓ Computer Networks.
- ✓ Embedded Systems

Topic-048. Software classification

Classification basics

- ✓ There are many different types of software performing different activities and have been developed for different tasks.
- ✓ Similar kind of software have been placed in one category.

Classification



Application Software

- ✓ Consists of the programs for performing tasks particular to the machine's utilization.
- ✓ spreadsheets, database systems, desktop publishing systems, accounting systems, program development software, and games.

Utility Software

- ✓ Consists of software units that extend (or perhaps customize) the capabilities of the operating system.
- ✓ eg. to format a magnetic disk, to copy a file from a magnetic disk to a CD
- ✓ Compress and decompress data
- ✓ Software for playing multimedia
- ✓ Handling network communication
- ✓ Application software and utility software can be vague.
- ✓ whether the package is part of the computer's "software infrastructure."
- ✓ once a project communicating over the internet was considered application software?

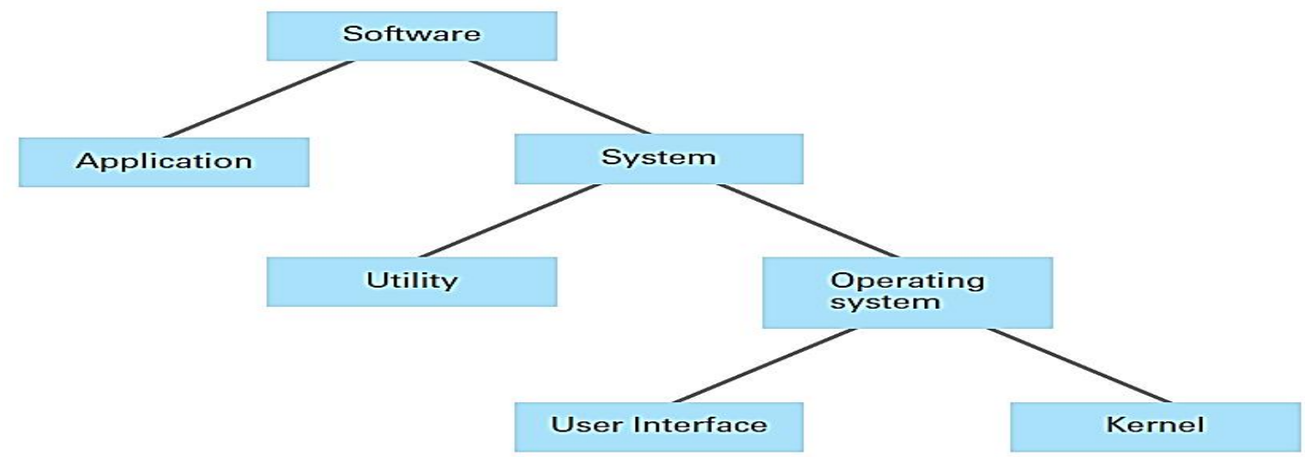
Summary

Software Classification

- ✓ Classification of Software
- ✓ Application Software
- ✓ Utility.

Topic-049. Component of OS-1

Software Classification



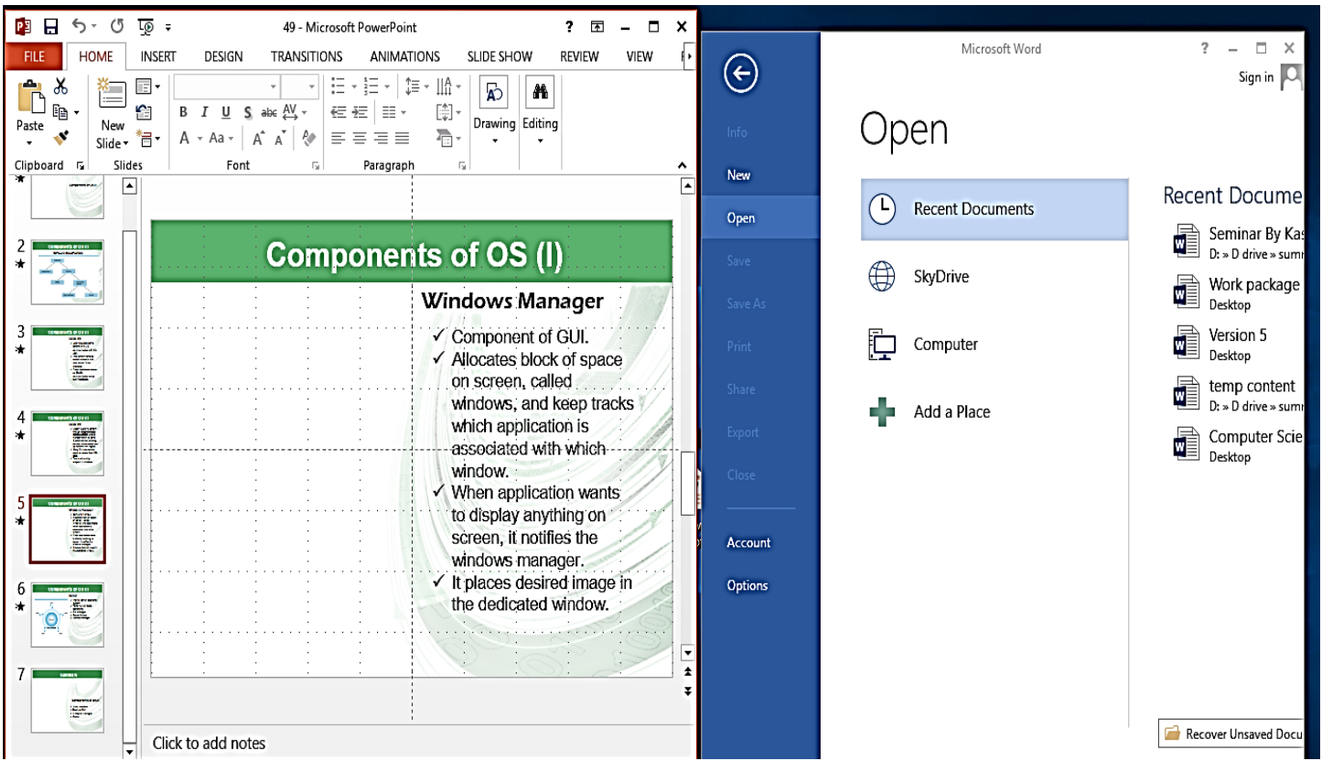
Inside OS

- ✓ User requests some actions and OS communicates with the user.
- ✓ The portion handles communication with user called "User Interface".
- ✓ Older interfaces known as **Shell** – communicates using text messages.
- ✓ Modern systems perform through **Graphical User Interface (GUI)**, pictorial representation as icons. Support mouse, pointing devices, touch screen can be handled with fingers.
- ✓ Early GUI applications could be loaded from MS-DOS.
- ✓ This is still a utility program in windows.

Windows Manager

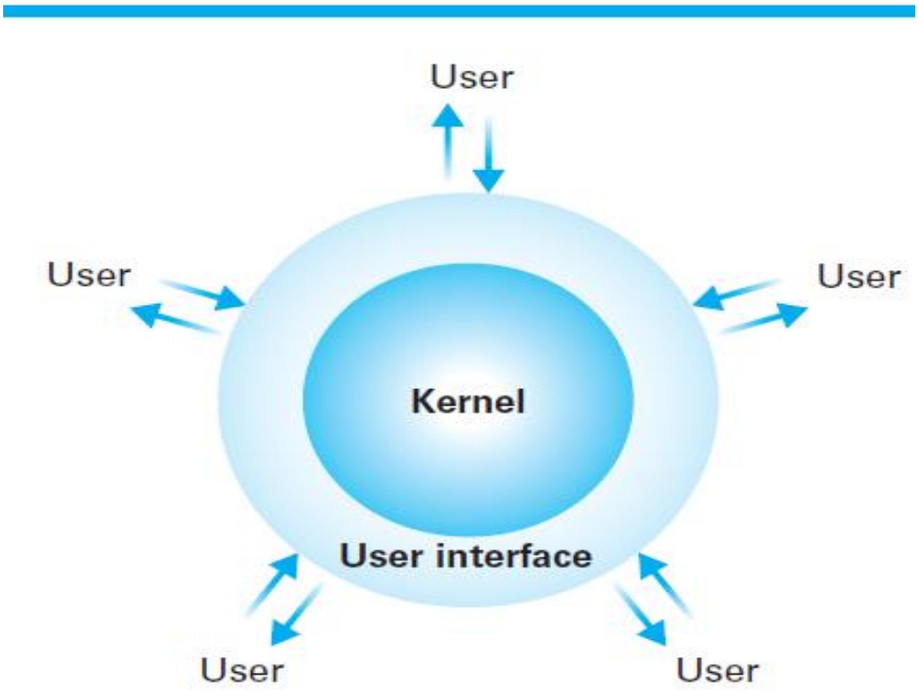
- ✓ Component of GUI.
- ✓ Allocates block of space on screen, called windows, and keep tracks which application is associated with which window.

- ✓ When application wants to display anything on screen, it notifies the windows manager.
- ✓ It places desired image in the dedicated window.



Kernel

- ✓ Internal part of Operating System.
- ✓ Performs very basic operations:
- ✓ File manager
- ✓ Device Drivers
- ✓ Memory manager



Components of OS (I)

- ✓ .User Interface
- ✓ Shell vs GUI
- ✓ Windows Manager
- ✓ Kernel

Topic-050. Component of OS-2

Kernel

- ✓ File manager
- ✓ Device Drivers

- ✓ Memory manager

File manager

- ✓ Coordinate the use of the machine's mass storage facilities.
- ✓ Where each file is located on mass storage,
- ✓ Which portion of mass storage is available
- ✓ Files can be grouped together into directory/folder.
- ✓ User can see directories files using windows explorer.

Device Drivers

- ✓ Software units that communicate with controllers or sometimes directly to peripheral devices.
- ✓ Translates the generic requests to specific steps.
- ✓ Generic OS can be customized for particular peripheral device by installing new driver.

Memory manager

- ✓ Coordination with main memory.
- ✓ Single user vs Multitasking needs:
- ✓ Which areas are free
- ✓ Which areas are occupied
- ✓ Which memory is assigned to which application.
- ✓ Assigning memory space
- ✓ Actions of each program are restricted to its own area.

Paging

- ✓ When available memory is less than the requirement, OS gets some space from mass storage and rotates data and program back and forth between main memory and mass storage called Paging.
- ✓ Uniform size pages.

Virtual Memory

- ✓ The memory taken from secondary storage during paging is termed as Virtual Memory.

Summary

Components of OS (II)

- ✓ .File Manager
- ✓ Device Drivers
- ✓ Memory Manager

Topic-051. Process of booting

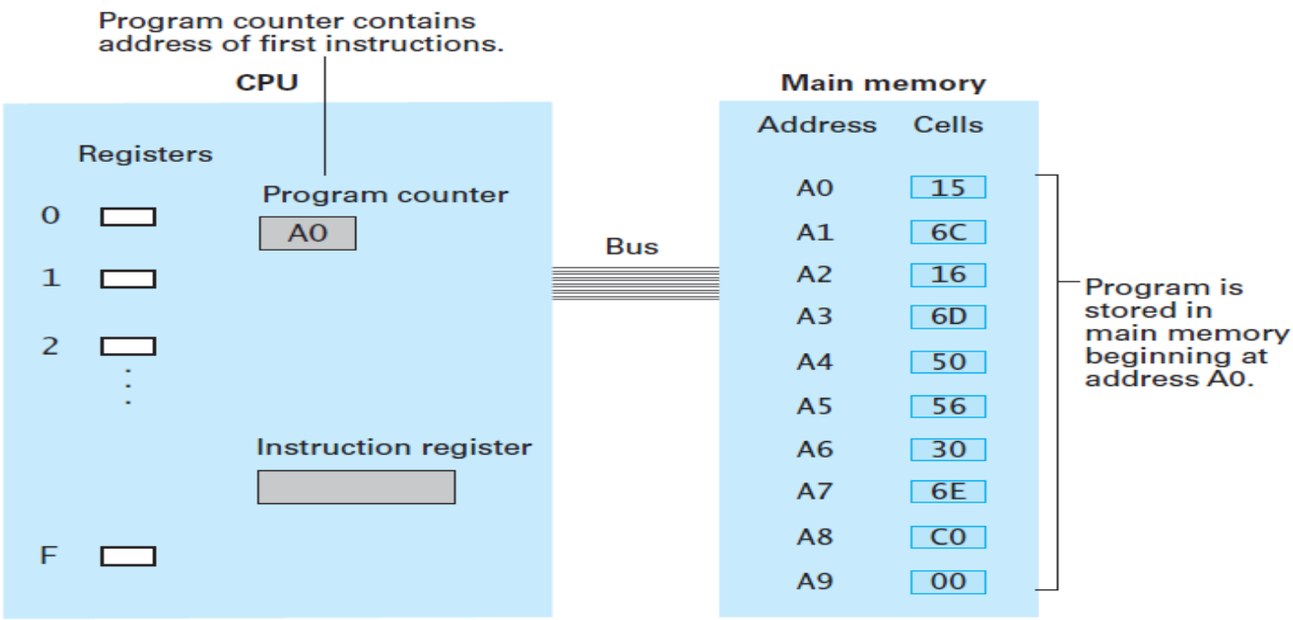
Why Booting?

- ✓ Where the OS is stored?
- ✓ Where the programs should be to be executed?

Booting

- ✓ Boot strapping
- ✓ Procedure that transfers the OS from mass storage to main memory.
- ✓ Main memory is empty when computer starts?
- ✓ How the OS will be loaded then?

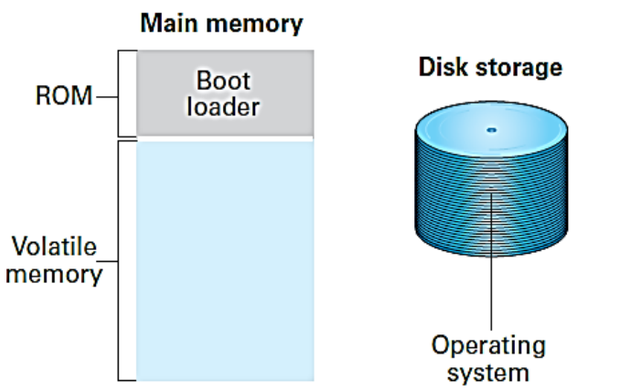
Recall Module 37



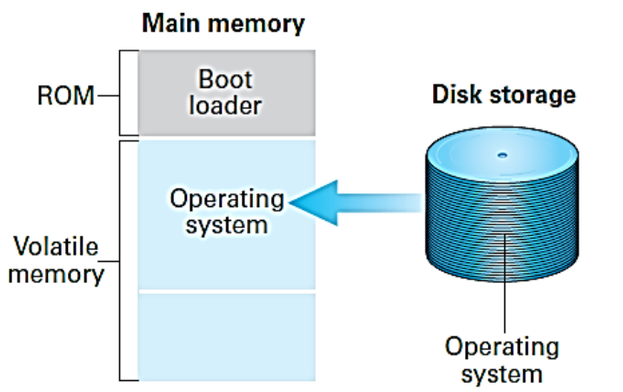
- ✓ Small portion of main memory is made from non-volatile memory cells where the CPU finds its way.
- ✓ ROM is read only memory, however, today it is constructed through flash memory so could be altered in special circumstances.

Boot Loader

- ✓ Permanently stored in ROM.
- ✓ Loads the OS into main memory.
- ✓ In smartphones, it can load from flash (nonvolatile) memory.
- ✓ In case of small workstations at large companies or universities, it can load from distant machine over network.



Step 1: Machine starts by executing the boot loader program already in memory. Operating system is stored in mass storage.



Step 2: Boot loader program directs the transfer of the operating system into main memory and then transfers control to it.

Summary

Process of Booting

- ✓ Why we need booting,
- ✓ How the OS is loaded in the main memory

Topic-052. Process and its administration-1

Coordinating Machine's activities

- ✓ How OS coordinates the execution of application software, utility software, or OS itself?

Program vs Process

- ✓ Fundamental concepts of modern operating systems is the distinction between a program and the activity of executing a program.
- ✓ Static vs dynamic activity.
- ✓ Analogy of music sheet.

Process

- ✓ The activity of executing a program under the control of the operating system is known as a **process**.
- ✓ Associated with a process is the current status of the activity, called the **process state**.

Process state

- ✓ The value of Program counter
- ✓ Values in other CPU registers
- ✓ Values in associated memory cell.
- ✓ Snapshot of the machine at particular time.

Summary

Process and its administration

- ✓ Distinction between process and program
- ✓ Process and its states

Topic-053. Process and its administration-2

Process Administration

- ✓ Scheduler and Dispatcher.
- ✓ **Scheduler** maintains the record which processes are present in the computer system,
- ✓ Introduction of new processes.
- ✓ Removal of completed processes.

Scheduler

- ✓ Maintains a process table in main memory.
- ✓ Process name, its ID, initiated by, priority, starting time, duration memory area (obtained from memory manager), current state etc.

Dispatcher

- ✓ Component of Kernel that oversees the process execution.
- ✓ Time sharing, multitasking, multiprogramming,
- ✓ Time Slice in mili or microseconds.

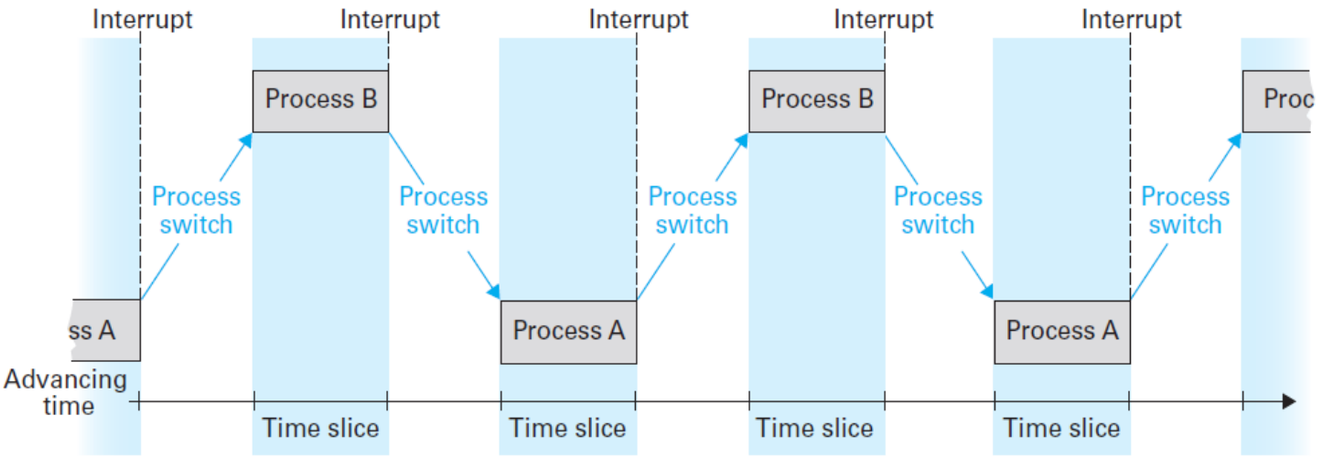
Process Switch

- ✓ Context Switch.
- ✓ Procedure of changing from one process to another.
- ✓ Dispatcher awards a time-slice, activates the timer circuit which generates an interrupt

Interrupt Handler

- ✓ When CPU receives an interrupt:
- ✓ Completes current machine cycle,
- ✓ Saves the current process
- ✓ Begin executing the program known as Interrupt handler.
- ✓ Part of dispatcher.
- ✓ Describer how the dispatcher should respond.
- ✓ Preempt the current process
- ✓ Transfer control back to dispatcher.
- ✓ Dispatcher selects the process from process table having highest priority from ready queue, time slice and timer circuit.

Multi-programming between Process A and process B



Process and its administration

- ✓ Scheduler
- ✓ Dispatcher
- ✓ Context Switch
- ✓ Interrupt Handler

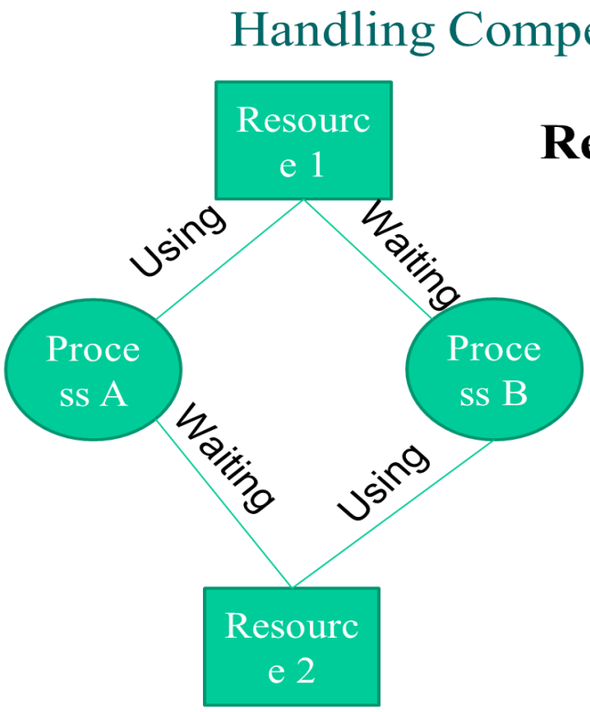
Topic-054. Handling competition between processes

Resource Allocation

- ✓ Important task of OS is to allocate the resources like peripheral devices, mass storage, main memory etc

Resources

- ✓ File manager allocates access to files
- ✓ Memory manager allocates the memory space.
- ✓ Scheduler allocates space in the process table
- ✓ Dispatcher allocates time-slots,
- ✓ Such resource allocation looks quite simpler, however, it's not so straight forward.
- ✓ Machine does not think about itself, it's You, who can/cannot design reliable OS



Resources Allocation

- ✓ Semaphores
- ✓ Deadlock

Summary

Handling competition in processes

- ✓ Resource
- ✓ Allocation
- ✓ Scenarios
- ✓ Semaphores
- ✓ Deadlocks

Topic-055. Semaphores

Handling Competition in Processes

Scenario

- ✓ Process A wants to use printer.
- ✓ OS must decide if the printer is available, then it must be given otherwise not?

How to track Printer Availability

- ✓ Using a single bit in memory, having states set (1) and clear (0)
- ✓ Flag clear (0) means printer is available
- ✓ Flag set (1) means printer is not available.

Issue in flag-implementation

- ✓ Set and clear is not executed in single instruction.
- ✓ Consider printer is available, 'Process A' reads this and before it could set the bit to 1, someone other reads the available zero bit.

Solution #1: disabling interrupts

- ✓ Disable interrupts before starting to execute the retrieve memory flag and enabling after it has been set as 1.

Solution #2: executing in one step

- ✓ **Test-and-set** instruction.
- ✓ CPU will complete this instruction and then will take next interrupt.
- ✓ A properly implemented flag is called **Semaphore**.

Critical Region

- ✓ Train track example
- ✓ Such a sequence of instruction is called Critical Region.
- ✓ The requirement that at one time one process can execute critical region is called mutual exclusion.

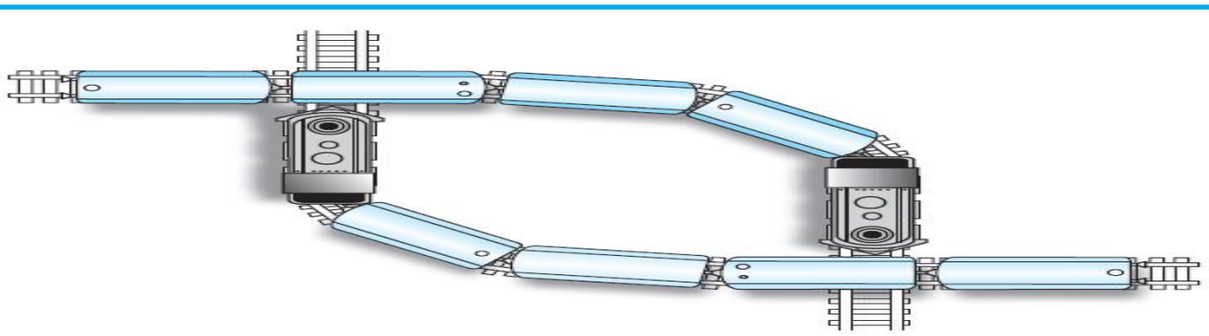
Summary

Semaphores

- ✓ Scenario
- ✓ How to track resource availability
- ✓ Issues
- ✓ Solutions
- ✓ Critical region
- ✓ Mutual exclusion

Topic-056. Deadlock

Deadlock



- ✓ “Process A” is using printer and wants CD-player, While “Process B” is using CD-player and wants printer.

Deadlock- another example

- ✓ Processes are allowed to create new space (**Forking** in UNIX)
- ✓ If scheduler has no space left in the process table, and each process must create another process before it can finish.

Deadlock- Conditions

1. There is competition for non-sharable resources.
2. The resources are requested on a partial basis; that is, having received some resources, a process will return later to request more.
3. Once a resource has been allocated, it cannot be forcibly retrieve

Deadlock Detection

- ✓ Identify deadlock when it has occurred and taking forcibly the resource from one process or **kill** some of the processes in process Table

Deadlock Avoidance

- ✓ Asking a process to request all resources at once – meeting the second condition.
- ✓ Or giving access to every time it asks – spooling.
- ✓ Involving controller to act as a device.

Summary

Deadlocks

- ✓ Examples
- ✓ Conditions
- ✓ Detection
- ✓ Avoidance
- ✓ Spooling

Topic-057. Security attacks form outside

Why Secure?

- ✓ As OS is overall in-charge of Computer System.
- ✓ Its role in maintaining security is important.
- ✓ Designing of dependable OS is vital

Security Issues

- ✓ **Reliability:** Flaw in the file manager could cause loss in the part of the file.
- ✓ **System Crash:** Defect in the dispatcher leads to system failure.
- ✓ Development of secure systems is not studied in OS, rather Software Engineering.

Stopping Unauthorized Persons

- ✓ Accounts for various types of users.
- ✓ User name, password, and privileges
- ✓ Login Procedure.
- ✓ Super-user/Administrator

Auditing Software

- ✓ Administrator can monitor activities.
- ✓ Malicious or accidental
- ✓ record and analyze the activities.

Auditing Software - Examples

- ✓ Flood of attempts to login using incorrect passwords
- ✓ Impossible Travel.
- ✓ Violating past behavior

- ✓ Presence of Sniffing Software.

Sniffing Software

- ✓ That records user activities and report them to intruder.
- ✓ An old example is fake simulation of login procedure of OS.

Major reasons of computer security

- ✓ Carelessness of users.
- ✓ Selection of user names and passwords.
- ✓ Sharing of passwords with friends
- ✓ Fail to change password on time.
- ✓ Import unapproved software.

Summary

Security Attacks from Outside

- ✓ Reliability & System crash
- ✓ Login
- ✓ Auditing Software
- ✓ Sniffing software
- ✓ Reasons of Computer security

Topic-058. Security attacks from inside

When malicious user is in:

- ✓ When an unauthorized access is gained:
- ✓ Look for information of interest.
- ✓ Places to insert destructive software.
- ✓ It is straightforward if administrator account has been hacked.

Using General User's Account:

- ✓ Trick the OS to grant unprivileged access:
- ✓ Trick memory manager to get access to those cells outside it's allotted area.
- ✓ Trick the file manager to access files whose access should be denied.
- ✓ Today's CPU are somehow smarter like giving access to only the allotted memory locations, otherwise OS would have been thrown away by some other process.
- ✓ Special purpose registers holds starting and ending addresses of processes.

Inherited Issue:

- ✓ Any process needs access outside the allotted area, can change the value of special purpose registers.

How to handle this:

- ✓ Privilege vs non-privilege mode:
- ✓ In non-privilege mode, limited number of instructions can be executed only.
- ✓ When system starts it's in privilege mode, when OS assign CPU to process, it is in non-privilege mode.

Security concerns:

- ✓ However, if it is compromised:
- ✓ A process can change time slot for itself to get more access on CPU.
- ✓ If process is allowed the access peripheral device, it can read/write without file manager supervision
- ✓ If process is allowed to access memory beyond its address, it can read write unauthenticated data.
- ✓ Maintaining security is important for administrator and for OS.

Summary

Security Attacks from inside

- ✓ Security threads.
- ✓ Administrator vs general user account
- ✓ Inherited issues
- ✓ Security threads

Topic-059. Network classification

Area-wise classification:

- ✓ Personal Area Network (PAN).
- ✓ Local Area Network (LAN)
- ✓ Metropolitan Area Network (MAN)
- ✓ Wide Area Network (WAN)

Personal Area Network (PAN)

- ✓ Short range communication (less than a few meters)
- ✓ Wireless headset and smartphone
- ✓ Wireless mouse and PC.

Local Area Network (LAN)

- ✓ Collection of computers in a single building.
- ✓ Computer in a university campus.
- ✓ Manufacturing plant.

Metropolitan Area Network (MAN)

- ✓ Network of intermediate size, one comprising of local community.
- ✓ Digital network based surveillance system of a city.
- ✓ Parking systems interconnected across the city.

Wide Area Network (WAN)

- ✓ Links machines over a greater distance in neighboring cities.
- ✓ Connection using internet

Open vs closed classification:

Open network: network's

internal operation is based on designs that are in the public domain. The Internet

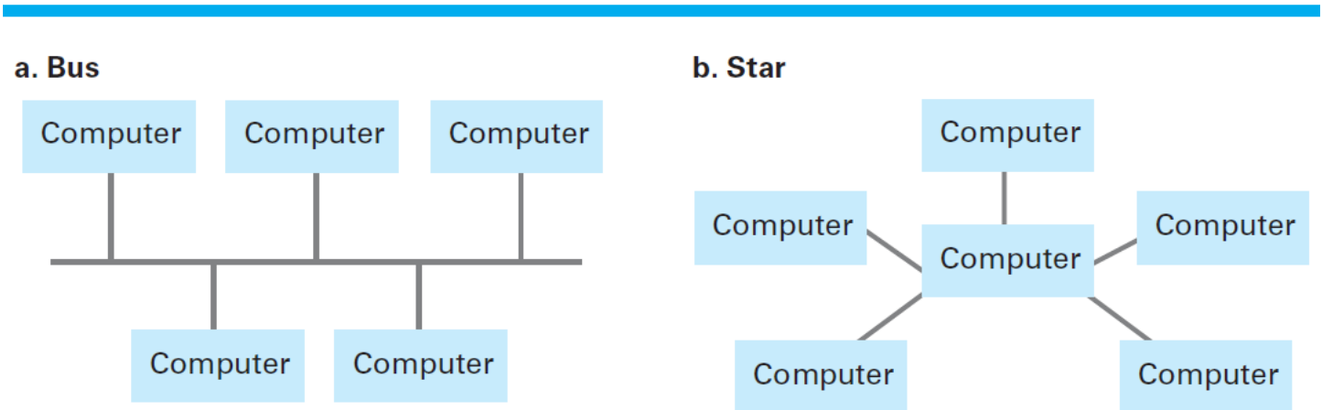
Closed network: innovations owned and controlled by a particular entity such as an individual or a

Corporation – **Security agencies**.

Topology based classification

- ✓ Topology of network which is pattern in which the machines are connected.
- ✓ **Bus:** connection through a common communication line called Bus.
- ✓ **Star:** one machine is central focal point and all other are connected to it.

Topologies



- ✓ Popular in 1990s when implemented in standard as Ethernet.
- ✓ Roots in 1970s, when one focal computer and many terminals

Summary

Network classification

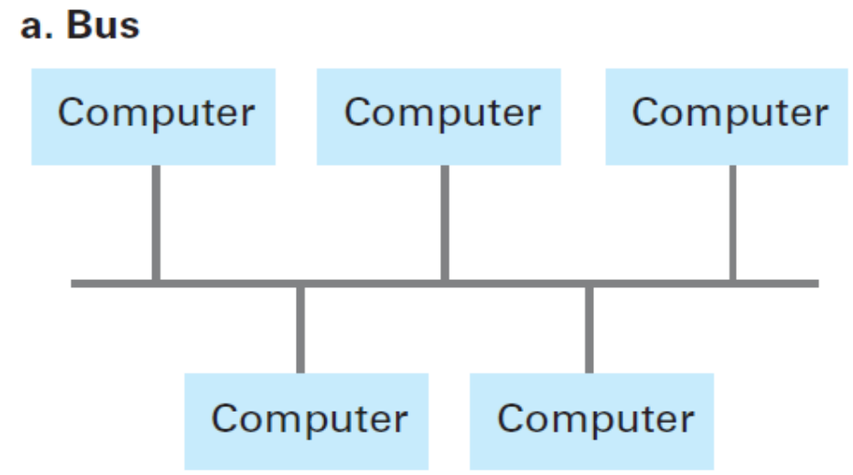
- ✓ Area based classification
- ✓ Open vs closed systems
- ✓ Topology based classification

Topic-060. Protocols

- ✓ Rules by which activities are conducted for a network to function reliably.

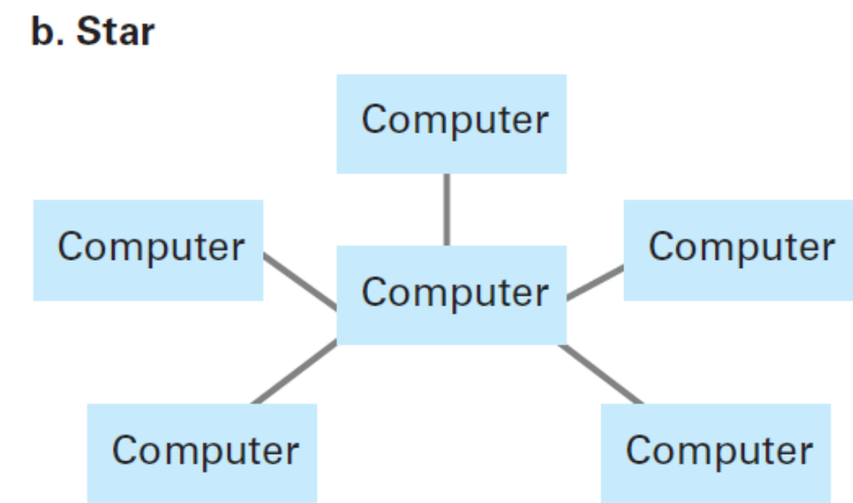
Benefit

- ✓ Vendors can make product for network applications compatible to other products.
  Scenario – Coordinating the transmission
- ✓ All might start transmission at once. Protocol Carrier Sense, Multiple Access with Collision Detection (**CSMA/CD**).

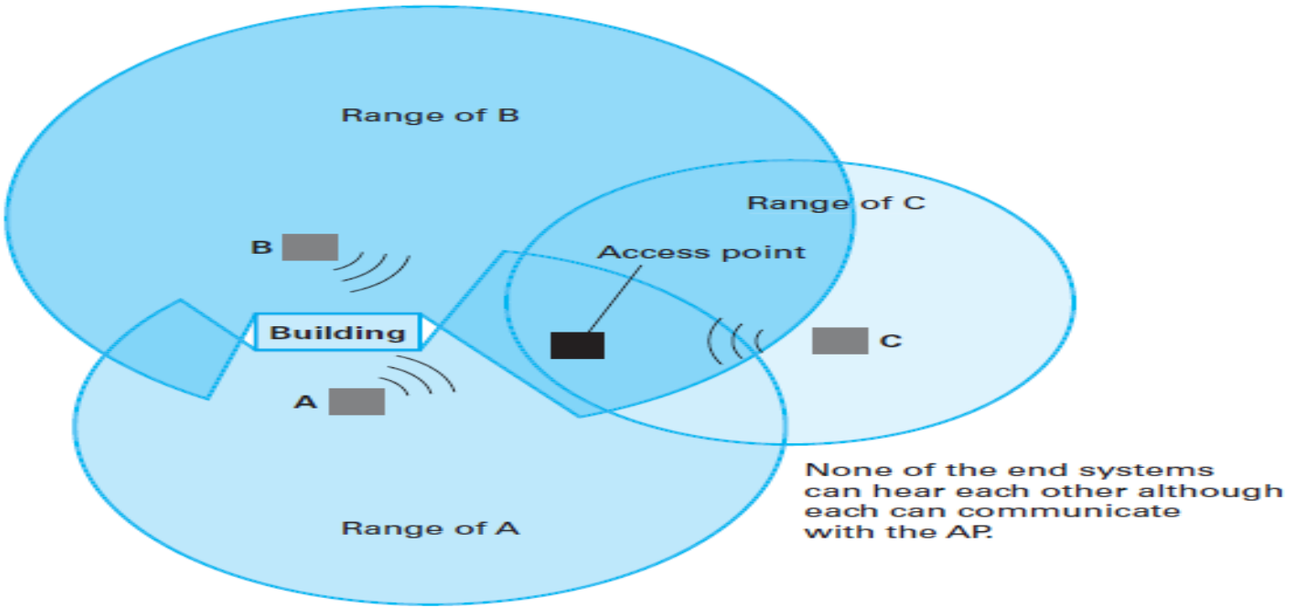


CSMA/CD

- ✓ Wait until the bus is silent,
- ✓ messages are broadcast, everyone keeps whatever it needs.
- ✓ If another machine also transmit, collision is detected, then wait before transmitting next packets.
- ✓ Difference with human conversation: Humans saying sorry, You please start, machines will start again.
- ✓ Not suitable for wireless star networks, it would not be possible to detect whether transmission is colliding:
- ✓ Due to signal drowns
- ✓ Or signals from different machines are blocked by objects or distance although they can communicate with the central AP known as Hidden Terminal problem.



Hidden Terminal Problem



Carrier Sense, Multiple Access with Collision Avoidance

- ✓ Due to hidden terminal problem, wireless networks adopt CSMA-CA.

CSMA-CA

- ✓ transmission starts when channel is free after short period of time, provided channel remains free in this period
- ✓ If machine has to wait, machine can claim a silent channel.
- ✓ At least no collision b/w new comers with old-ones.
- ✓ Hidden terminal problem is not solved even with this.

Summary

Protocol

- ✓ Rules for transmission
- ✓ CSMA-CD
- ✓ Hidden Terminal Problem
- ✓ CSMA-CA

Topic-061. Combining networks

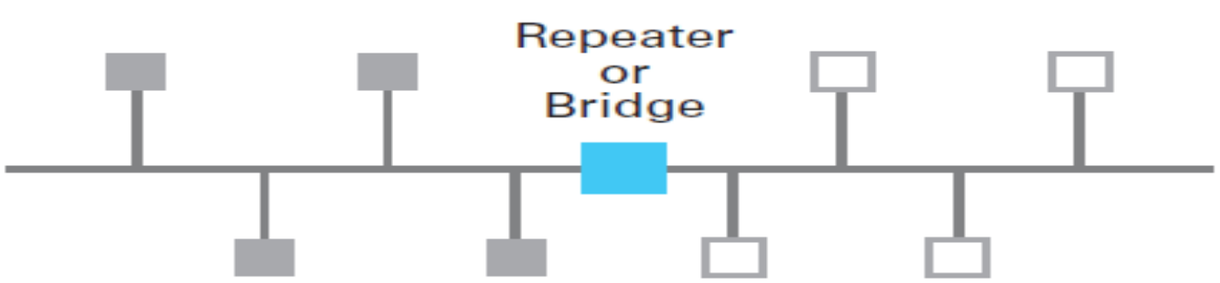
Why to Combine:

- ✓ Necessary to connect existing networks to form an extended communication system.
- ✓ Same type of network can be combined to form a bigger network.

Options:

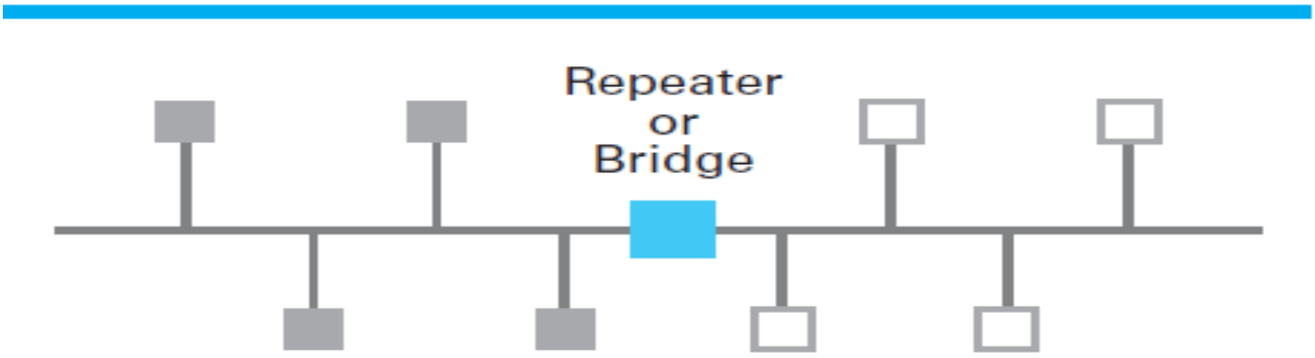
- ✓ Repeaters.
- ✓ Bridges
- ✓ Switches

Repeaters



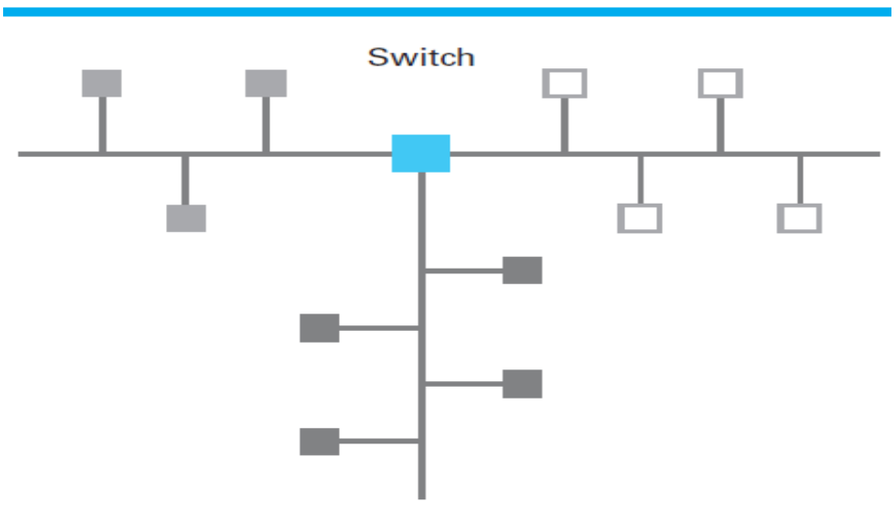
- ✓ Device that passes signals back and forth b/w two original buses with some form of amplification without understanding the meanings.

Bridge



Switch

- ✓ Connects several buses rather than just two.
- ✓ Works like bridge, forwards only relevant messages

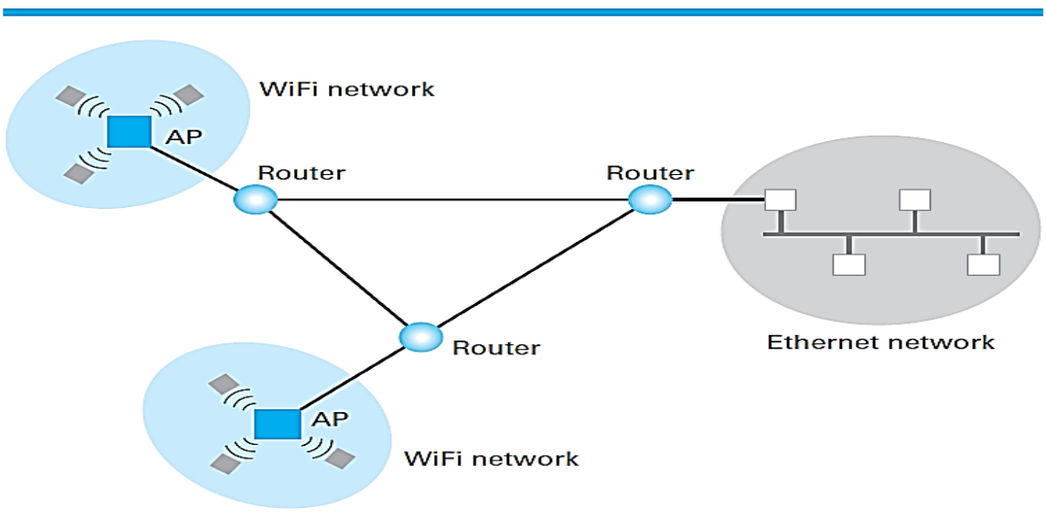


Limitation?

- ✓ When we have more than one type of network, we can not use: repeaters, bridge, or switch

Router

- ✓ Special computers to forward message over diversified characterized networks: internet.
- ✓ Provides links.
- ✓ Routers communicates and help to transfer message



Router

- ✓ Maintains the **Forwarding Table:** directions in which messages should be sent based on destination.
- ✓ **Gateway:** The “point” at which one network is linked to an internet

Summary

Combining Networks

- ✓ Repeaters
- ✓ Bridges
- ✓ Switches
- ✓ Routers
- ✓ Forwarding Table
- ✓ Gateway

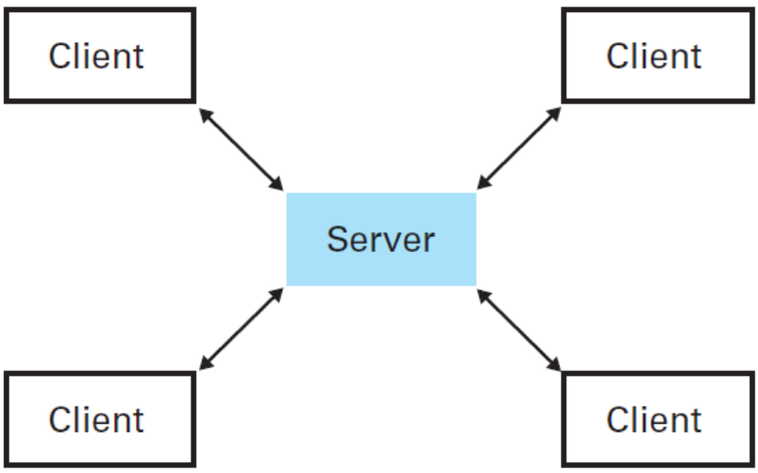
Topic-062. Methods of process communication

Inter-process Communication:

- ✓ Different processes on network or even in time-sharing fashion on single machine need to communicate with each other.

Client/server model

- ✓ One of the popular convention.
- ✓ One machine is acting as server, others as client.
- ✓ Client is making request and server is satisfying the request



Client/server model

Example 1

- ✓ An early application using client/server model was a central printer.
- ✓ Printer (**Print Server**) acts as server and all office machines as clients.

Example 2

- ✓ To reduce the cost of magnetic disk and to reduce the cost of duplicate file storage on all machines in an organization, a central server of files (**File Server**) acts as server and all machine requests to retrieve and store files.

Peer-to-peer (P2P)

- ✓ the peer-to-peer model involves processes that provide service to and receive service from each other.
- ✓ Instant messaging
- ✓ Competitive interactive games

- ✓ Distributing files like music recordings, motion pictures.
- ✓ The collection of peers in such distribution is sometimes called **Swarm**.

Why P2P ?

- ✓ Decentralized operations give more efficiency.

Summary

Methods of process communication

- ✓ Client/Server
- ✓ Peer-to-peer

Topic-063. Distributed systems

- ✓ Software units that are executed as processes on different computers.

Which software are distributed?

- ✓ With Networking technologies success:
- ✓ Global information retrieval, company wide accounting and inventory, computer games, even the software controlling the network infrastructure are designed distributed.

Evolution

- ✓ Early DS were built from scratch.
- ✓ However, now a days, basic infrastructure including communication and security systems are available on which you need to build only unique part of application.

Types of DS

- ✓ Cluster Computing.
- ✓ Grid Computing.
- ✓ Cloud Computing.

Cluster Computing

- ✓ Many Independent computers work closely together to provide computation and services.
- ✓ Cost of machines + cost of high speed network remains less than the cost of higher speed supercomputer.

Advantages of Cluster Computing

- ✓ Low maintenance cost.
- ✓ Provide high-availability.
- ✓ Load balancing

Grid Computing

- ✓ loosely coupled than clusters but that still work together to accomplish large tasks.
- ✓ involve specialized software to make it easier to distribute data and algorithms to the machines eg. University of Wisconsin's **Condor system** or **Berkeley's Open Infrastructure for Network Computing (BOINC).**
- ✓ Software are installed for volunteer computing.
- ✓ When the machine is free, it can join the grid, power of million of computers to solve complex large problems

Cloud Computing

- ✓ Huge pools of shared computers on the network can be allocated for use by clients as needed.
- ✓ Amazon's Elastic Compute Cloud allow clients to rent virtual computers by the hour
- ✓ Google Drive and Google Apps.
- ✓ Reasonable guarantees of reliability and scalability.
- ✓ Concerns about privacy and security, when we may no longer know who owns and operates the computers that we use.

Summary

Distributed Computing

- ✓ Cluster Computing
- ✓ Cloud Computing
- ✓ Grid Computing

Topic-064. Internet architecture

internet and Internet

- ✓ internet is network of networks of computers, normally each network following different topology.
- ✓ Internet is a specific and world wide internet.
- ✓ Here we will discuss the Internet Architecture.

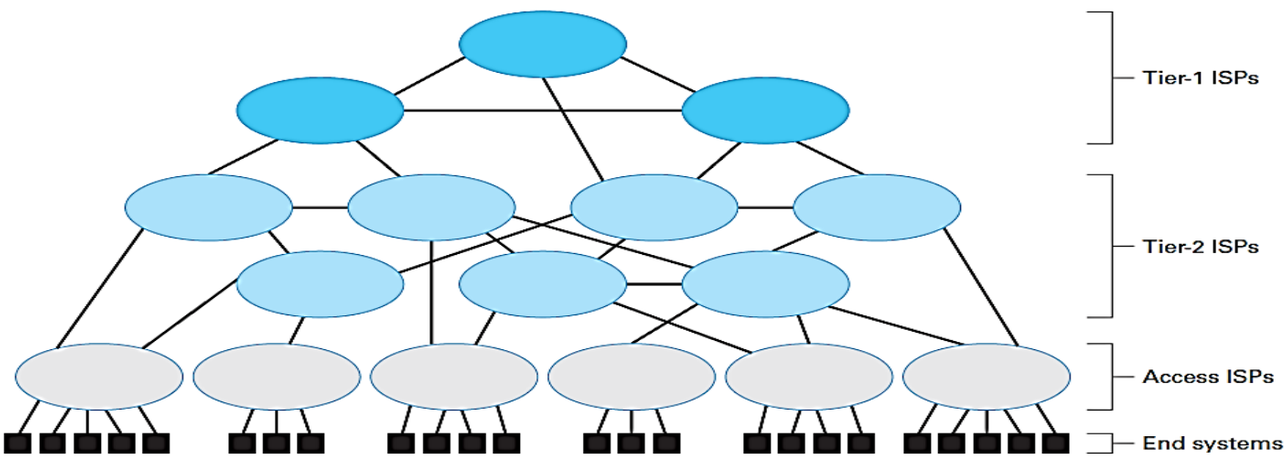
Internet

- ✓ Started as a research efforts in 1960's.
- ✓ The goal was to link variety of networks to form a connected functional large network, not disrupted by local disaster.
- ✓ Started as government project by Defense Advanced Research Projects Agency (DAR-Pa)
- ✓ Shifted from government sponsored project to academic research project.
- ✓ Today it's a commercial undertaking linking PANs, LANs, MANS, and WANs involving million of computers.

Internet Architecture

- ✓ Individual networks are managed by Internet Service Providers (**ISPs**).

Internet composition



Tier-1 ISP

- ✓ Relatively few high speed, high capacity, international WANs
- ✓ Thought of as backbone of the Internet.
- ✓ Operated by large companies normally originated as telephone company and extended.

Tier-2 ISP

- ✓ Connecting with Tier-1 ISPs.
- ✓ More regional in scope, relatively less strong than Tier-1.
- ✓ Tier-1 and Tier-2 are network of routers
- ✓ Forming the core of the Internet.

Tier-3/Access ISP

- ✓ Independent internet, sometimes referred as intranet, operated by a single authority.
- ✓ Eg. Cable and telephone companies.
- ✓ The devices connecting to access ISPs are end systems or hosts.

End systems or Hosts

- ✓ Laptops/PCs.
- ✓ Telephones
- ✓ Video Cameras
- ✓ Automobiles
- ✓ Home appliances
- ✓ The area within the AP or group of APs' range is often called a hot spot.

Summary

Internet Architecture

- ✓ internet vs Internet
- ✓ Internet Architecture
- ✓ Tier-1, Tier-2, and Tier-3, end systems

Topic-065. Internet addressing

Why Addressing

- ✓ How we call humans- their names.
- ✓ Ambiguities in names?
- ✓ Each computer should have its own identity (Unique name).
- ✓ Such unique address is known as IP address,
- ✓ IP (Internet Protocol)

Bits to represent addresses

- ✓ Originally 32 bits were assigned to mention any address. You know maximum 2^{32} can be addressed.
- ✓ Now effort in IPV6 is undergoing to convert addresses to 128 bits

organization

- ✓ Internet Cooperation for Assigned Names and Numbers (**ICANN**)
- ✓ Non profit organization

Dotted decimal notation

- ✓ Bytes of address are separated by periods.
- ✓ Each byte is expressed in traditional base 10 notation.
- ✓ Like 5.2 would be represented as
- ✓ 0000010100000010
- ✓ Practice on power method.

Still difficult for human to remember?

- ✓ Machines are identified as mnemonic names.
- ✓ Concept of **domain** (thought of as regions)
- ✓ Regions are not geographical regions.
- ✓ University (.edu), organization (.org), government (.gov)

Domain name registration

- ✓ With ICANN , the process of registration by companies is known as **Registrars**.
- ✓ vu.edu.pk.
- ✓ .edu, .org, .gov are called “T**op level domains”**.
- ✓ .pk for Pakistan, .uk for United Kingdom are called “**Country Level TLDs**”

Subdomains

- ✓ When the domain is registered, subdomains can be formed by the organization its self, should be unique like:
- ✓ ocw.vu.edu.pk
- ✓ Multiple subdomains may be formed like
- ✓ abc.ocw.**vu.edu.pk**

Domain to Address conversion

- ✓ Domains were easier to understand by humans, however, communication will be done via IP addresses.
- ✓ Such conversion is done using **name servers**.
- ✓ These name servers use international directory: **Domain name server.**
- ✓ Process of using DNS is called **DNS-Lookup.**

Summary

Internet Addressing

- ✓ Why Addressing
- ✓ ICANN
- ✓ Dotted Decimal Notation
- ✓ Domain names
- ✓ Name Servers,
- ✓ Domain Name Servers
- ✓ DNS-Lookup

Topic-066. Internet applications

Early network based Applications

- ✓ Newsreader application
- ✓ File transfer applications
- ✓ Accessing another computer from great distance

Newsreader applications

- ✓ Contacted servers using a protocol known as: **"Network News Transfer Protocol (NNTP)".**

File Transfer applications

- ✓ Listing and copying files across network, **File Transfer protocol (FTP).**

Accessing remote Computer

- ✓ **Telnet** protocol or later the secured shell.

HTTP

- ✓ Today, webservers and browsers are more sophisticated, now these applications and many more applications are being handled with one common protocol known as: **Hyper Text Transfer Protocol (HTTP)**

Applications

- ✓ Emails
- ✓ Voice over IP (VOIP)
- ✓ Internet Multimedia Streaming

Summary

Internet Applications

- ✓ Different protocols for different applications
- ✓ HTTP

Topic-067. Internet application: email

Messaging applications

- ✓ Instant Messaging
- ✓ Browser based chatting
- ✓ Twitter based tweets
- ✓ Facebook wall.
- ✓ One of the oldest is Electronic mail **(Email)**

Messaging applications

- ✓ Simple Mail Transfer Protocol **(SMTP)**
- ✓ **Scenario:**

Scenario

- ✓ mafzal from cust.edu.pk wants to send email to hmaurer from iicm.tugraz.at

Internet Applications: Email

1 220 mail.iicm.tugraz.at SMTP Sendmail Gallifrey-1.0; Sun, 12

Aug 2018 14:34:10

2 HELO mail.cust.edu.pk

3 250 mail.iicm.tugraz.at Hello mail.cust.edu.pk, pleased to meet you

4 MAIL From: mafzal@cust.edu.pk

5 250 2.1.0 mafzal@cust.edu.pk... Sender ok

6 RCPT To: hmaurer@iicm.tugraz.at

7 250 2.1.5 hmaurer@iicm.tugraz.at... Recipient ok

8 DATA

9 354 Enter mail, end with "." on a line by itself

10 Subject: Extermination.

12 EXTERMINATE!

13 Regards, Dr. M. Tanvir Afzal

15 250 2.0.0 r7NJYAEl028071 Message accepted for delivery

16 QUIT

17 221 2.0.0 mail.iicm.tugraz.at closing connection

Other Protocols

- ✓ SMTP for text messages.
- ✓ MIME (Multipurpose Internet Mail Extensions) to convert non-ASCII to SMTP compatible form.

Other Protocols

- ✓ Post Office Protocol Version 3 (POP3)
- ✓ Internet Mail Access Protocol (IMAP)
- ✓ User can download and maintain mail messages into folders etc, POP3 helps to store messages on local machine, IMAP on mail server machine.

Summary

Internet Applications: Email

- ✓ SMTP
- ✓ Working Scenario
- ✓ MIME
- ✓ POP3
- ✓ IMAP

Topic-068. Internet application: voip

What is VOIP

- ✓ Voice over Internet Protocol (VOIP).
- ✓ Internet infrastructure is used to provide voice communication similar to that of traditional telephone systems

VOIP consists of:

- ✓ Two processes on different systems using p2p model.
- ✓ No significant issues in p2p as you might remember

VOIP Challenges:

- ✓ How to initiate and receive call
- ✓ Linking VOIP with traditional telephones.
- ✓ Services like 1122, 15.

- ✓ Companies consider it as threat to their business, and have made sure to tax them heavily from government or outlawed it completely.

Existing VOIP Systems:

- ✓ VOIP Soft Phones.
- ✓ Analog Telephone Adapters
- ✓ Embedded VOIP phones
- ✓ Wireless VOIP Technology.

VOIP Soft Phones:

- ✓ P2P software on both computers without any special hardware.
- ✓ Skype
- ✓ Also provide links with traditional telephone communication system.
- ✓ Drawback: its proprietary, operational standards are not publically known.

Analog Telephone Adapters:

- ✓ devices that allow a user to connect his or her traditional telephone to phone service provided by an access ISP.
- ✓ choice is frequently bundled with traditional internet service and/or digital television service

Embedded VOIP Phones:

- ✓ Devices that replace a traditional telephone with an equivalent handset connected directly to a TCP/IP network

Wireless VOIP Technology:

- ✓ Earlier generations of wireless phones use company's network, their protocol, access through gateway between company's network and Internet.
- ✓ Now 4G phone network is an IP-based network throughout

Summary

Internet Applications: VOIP

- ✓ Challenges.
- ✓ VOIP Soft Phones.
- ✓ Analog Telephone Adapters
- ✓ Embedded VOIP phones
- ✓ Wireless VOIP Technology.

Topic-069. Internet multimedia streaming

Streaming

- ✓ Transporting audio and video across the Internet in real-time is known as **streaming.**
- ✓ **Netflix** claimed in 2015 that 42.5 billion hours videos have been streamlined by users.
- ✓ In first quarter of 2018, 7.5 million new users have been added.
- ✓ Netflix and Youtube consumed half of Internet Bandwidth.

Content Distribution Strategies

- ✓ Unicast
- ✓ N-unicast.
- ✓ Multicast
- ✓ On-demand Streaming

Unicast

- ✓ Internet Radio Station
- ✓ One sender is sending to one receiver.

N-unicast

- ✓ Internet Radio Station
- ✓ Single sender involved with multiple unicasts.
- ✓ Challenging for server and its neighboring systems as they should also help to send to required clients.

P2P

- ✓ Once a peer has received a data, that peer sends data to next peer.
- ✓ Distribution load has been distributed b/w peers rather than server as was in N-unicast.

Multicast

- ✓ Distribution problem to the Internet routers.
- ✓ Server transmit a message to multiple clients by relying on routers.
- ✓ Functionality of Routers has been extended beyond their normal work load.
- ✓ Has been implemented in small networks, not still implemented for the Internet.

On demand Streaming

- ✓ End user view multimedia on their own time, preferences, rewind, forward.
- ✓ To service million of users, Content Delivery Networks (CDN) have been formed.
- ✓ Group of servers distributed strategically

Content Delivery Networks

- ✓ Helps to support the streaming task to neighboring devices.
- ✓ most of the cases, this is available in Access ISP.
- ✓ **Anycast:** enables an end user to automatically connect to the closest server

Summary

Internet Multimedia Streaming

- ✓ Streaming
- ✓ Unicast
- ✓ N-unicast.
- ✓ Multicast
- ✓ On-demand Streaming
- ✓ Content Delivery Networks
- ✓ Anycast

Topic-070. World wide web

Idea of Web

- ✓ While working at CERN, Tim Berners Lee and his team, notably Robert Cailliau submitted a proposal to manage documents
- ✓ They wanted to link the physics documents of CERN.

Contemporary Proposals

- ✓ First Gopher Servers (Mark McCahill): 1991
- ✓ First WWW Server (CERN): 1991
- ✓ First Hyper-G Applications (Graz): 1991
- ✓ Breakthrough of WWW due to first graphic browser Mosaic in 1993.

Gopher widespread

- ✓ In 1993, Gopher was most widespread all over the world (80,000 servers), WWW and Hyperwave had below 100 each. But situation started to turn in favour of WWW in the same year.

Important features missed out in WWW

- ✓ Version control.
- ✓ Transclusions
- ✓ Access management
- ✓ Avoid broken links
- ✓ Store metadata
- ✓ Flexible search
- ✓ Annotations to everything

Basic Terms

- ✓ Hypertext – termed defined by Ted Nelson 1965 - Linked Documents
- ✓ Hyperlinks to link hypertext documents.
- ✓ http://vu.edu.pk/

Summary

World Wide Web

- ✓ History of Web

- ✓ Contemporary systems
- ✓ Important features missed out in Web
- ✓ Hypertext document
- ✓ Hyperlinks

Topic-071. Web implementations

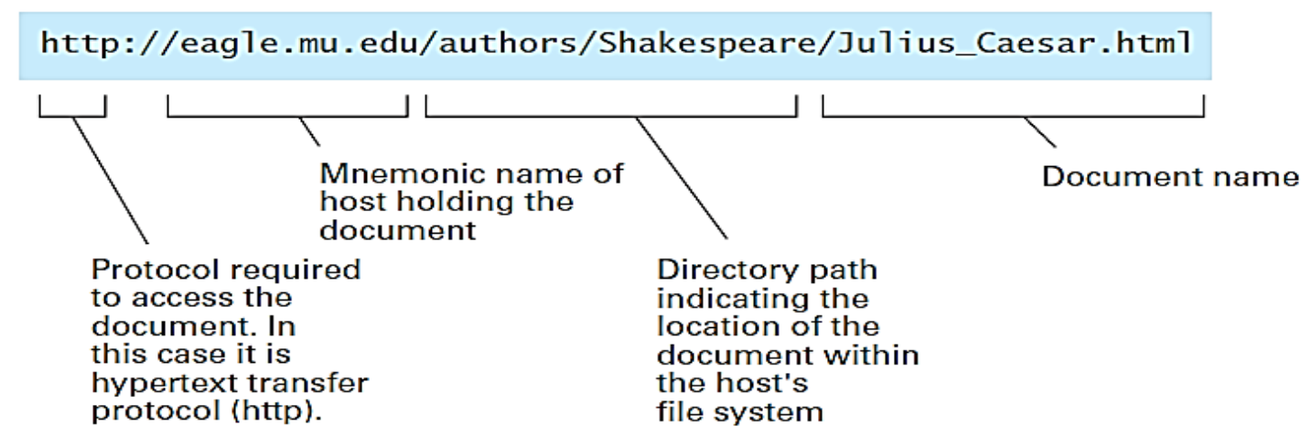
Accessing Internet

- ✓ **Browsers:** resides on user computer, obtains the material requested by the user from the web server.
- ✓ **Web servers:** resides on a computer containing hypertext documents to be accessed.
- ✓ Documents are transferred using HTTP.

URL

- ✓ Each document on the web is given a unique address known as Uniform Resource Locator **(URL).**
- ✓ Each URL has all information to access the document.

Example of an URL



Incomplete URL

- ✓ http://www.vu.edu.pk
- ✓ We have not given directory path and document name
- ✓ Homepage
- ✓ What if www.vu.edu.pk is given.

Summary

Web implementations

- ✓ Web browsers
- ✓ Web servers
- ✓ URL

Topic-072. HTML

What is HTML?

- ✓ Similar to text document, however, just contains some **tags**.
- ✓ Tags: describe how the content should appear on the screen.
- ✓ This system of tags is known as HTML (**Hypertext Markup Language**)

Web browsers

- ✓ The information written in tags is then processed by web browsers – how to display the information.
- ✓ Lets see some examples:
- ✓ vu.edu.pk

- ✓ hec.gov.pk

Basic structure of HTML document

```
<html>
<head>
<title>demonstration page</title>
</head>
<body>
<h1>Hello</h1>
<p>Introduction to Computing.</p>
</body>
</html>
```

Summary

HTML

- ✓ What is HTML
- ✓ What are Tags
- ✓ View Source
- ✓ Examples of different HTML pages.
- ✓ HTML basic structure

Topic-073. Making simple HTML page

HTML Structure?

```
<html>
<head>
<title>demonstration page</title>
</head>
<body>
<h1>My Web Page</h1>
<p>Click here for another page.</p>
</body>
</html>
```

Lets do it practically

Summary

HTML

- ✓ Making simple web page
- ✓ Using anchor tag <a>

Topic-074. More on HTML

Self Learning

W3schools
https://www.w3schools.com/tags/default.asp

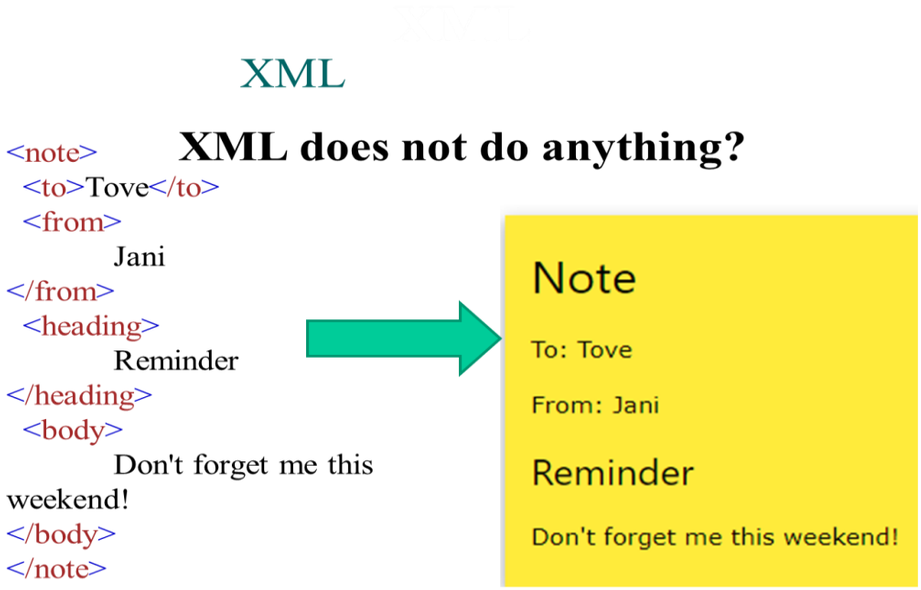
More on HTML

- ✓ w3Schools

Topic-075. XML

What is XML?

- ✓ XML stands for eXtensible Markup Language
- ✓ Much like HTML
- ✓ Was designed to store and transport data
- ✓ Self Descriptive



XML vs HTML?

- ✓ XML was designed to carry data
- ✓ HTML was designed to display data.
- ✓ XML tags are not predefined as HTML.
- ✓ You can choose tags as per your requirement and convenience
- ✓ Author will define the tags and data.

Example

- ✓ Lets suppose we want to library data in an xml file, we have the following information abut each book:
  - ✓ Book category
  - ✓ Title
  - ✓ Author
  - ✓ Year
  - ✓ Price

Example – XML file

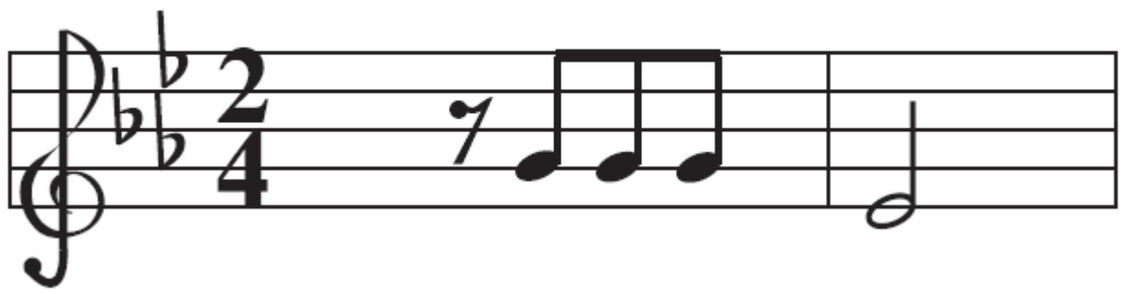
- ✓ <title>, <author>, <year>, and <price> have **text content** because they contain text (like 29.99).
- ✓ <bookstore> and <book> have **element contents**, because they contain elements.
- ✓ <book> has an **attribute** (category="children").

```
<bookstore>
 <book category="children">
  <title>Harry Potter</title>
```

```
    <author>J K. Rowling</author>
    <year>2005</year>
    <price>29.99</price>
  </book>
  <book category="web">
    <title>Learning XML</title>
    <author>Erik T. Ray</author>
    <year>2003</year>
    <price>39.95</price>
  </book>
</bookstore>
```

Another Example

```
<staff clef = "treble">
<key>C minor</key>
<time> 2/4 </time>
<measure> <rest> egth </rest> <notes> egth G, egth G, egth G </notes></measure>
<measure> <notes> hlf E </notes></measure>
</staff>
```



XML benefits

- ✓ It simplifies data sharing
- ✓ It simplifies data transport
- ✓ It simplifies platform changes
- ✓ It simplifies data availability.

Summary

XML

- ✓ What is XML
- ✓ Examples
- ✓ Why we need it?
- ✓ benefits

Topic-076. Client side and server side

How your browser works?

- ✓ You type a URL
- ✓ Web browser will contact the server to give back the latest page as source file.
- ✓ That source file will be understood and rendered by your browser to display your intended page.

When if?

- ✓ The page contained some animation.
- ✓ The feedback form need to be filled by the user.

- ✓ Some additional activities need to be performed.
- ✓ **Client-side:** such activities are performed by client (browser)
- ✓ **Server-side:** such activities are performed by servers.

Examples

- ✓ User making queries at search engines.
- ✓ Email reading and writing

Controlling Client-side activities

- ✓ **JavaScript** developed by Netscape Communications Inc. within HTML.
- ✓ Transfer webpage to browser and then transfer additional units called **Applets** as requested by HTML document, developed by Sun Microsystems.
- ✓ Macromedia Flash, multimedia client side presentations are managed.

Controlling server-side activities

- ✓ Early technology was **CGI (Common Gateway Interface)**, client could request servers.
- ✓ A variation by Sun Microsystem is **Servlets, Java Server Pages (JSP)**
- ✓ A similar approach by Microsoft is known as **Active Server Pages (ASP)**
- ✓ Open source system: **PHP Hypertext Preprocessors.**

Ethical and Security concerns

- ✓ Whatever sent by server may not be executed by client blindly and vice versa.
- ✓ This could leads to security and ethical concerns.

Summary

Client-side and Server-side

- ✓ How browser renders the web pages.
- ✓ Examples
- ✓ Approaches to handle client-side activities.
- ✓ Approaches to handle server-side activities.
- ✓ Security and ethical issues.

Topic-077. Layered approach to internet software-i

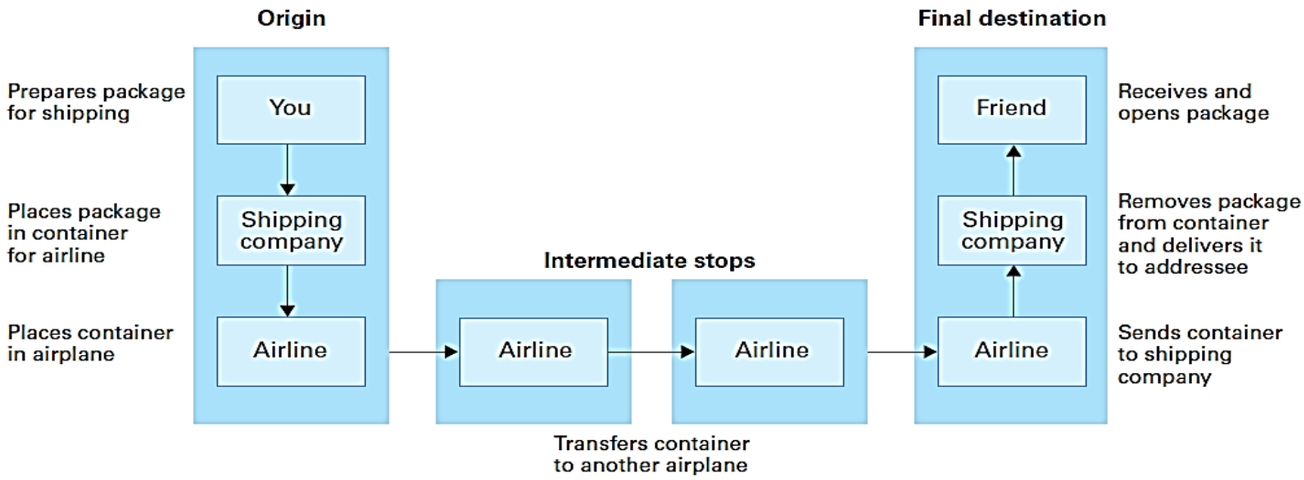
Networking Software

- ✓ Networking software provide infrastructure to transfer messages.
- ✓ This activity is executed in hierarchy of software units.

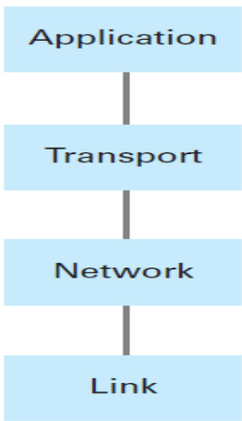
Lets consider a scenario?

- ✓ You are living in Lahore and want to send some gift to your friend living in Switzerland.
- ✓ Lets see what would be the procedure?

Scenario execution



Network Communication



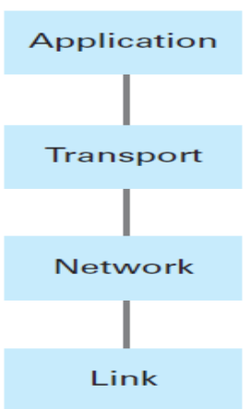
Summary

Layered Approach to Internet Software I

- ✓ Networking Software
- ✓ Scenario Gift sending
- ✓ Preliminary discussion Four stage model.

Topic-078. Layered approach to internet software-ii

Network Communication



Application Layer

- ✓ Software like Clients and Servers.
- ✓ Utility packages
- ✓ FTP
- ✓ Remote login
- ✓ Uses transport as you used shipping company.

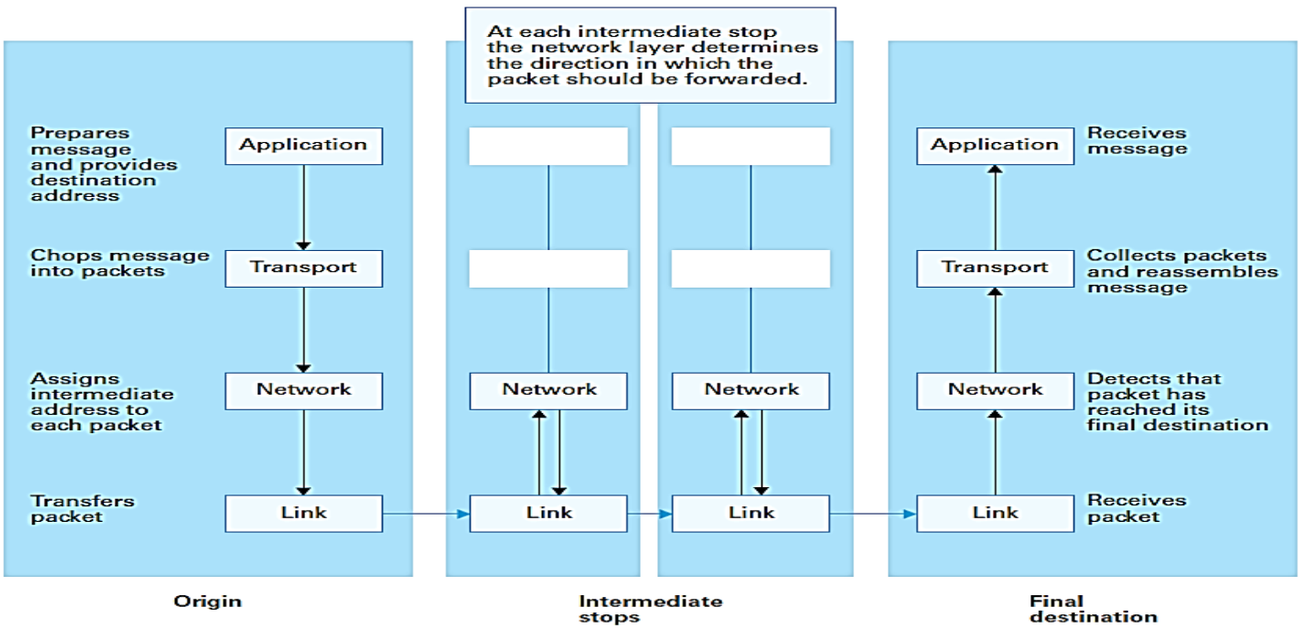
- ✓ Correct address using name servers is also its responsibility

Transport Layer

- ✓ Receives messages from Application layer
- ✓ Checks whether properly formatted
- ✓ And transmission over the Internet.
- ✓ Divides long messages to short
- ✓ Assign them numbers
- ✓ Hands-over **Packets** to network layer

Network Layer

- ✓ Decides the direction to which the packet should move.
- ✓ Combination of network layer and link layer form a software residing at routers.
- ✓ Link layer is in charge to send and receive the packets.



Summary

Layered Approach to Internet Software II

- ✓ Application Layer.
- ✓ Transport Layer.
- ✓ Network Layer.
- ✓ Link Layer.
- ✓ Scenario

Topic-079. Layered approach to internet software-iii

Network Layer

- ✓ When packet is received, it uses forwarding table to send the packet.
- ✓ Hands-in the packet to Link layer.
- ✓ Link layer then process the packet to its destination

Link Layer

- ✓ If the network is Ethernet, it uses CSMA/CD
- ✓ If its wifi, it uses CSMA/CA.
- ✓ When the packet is received to link layer of other side, it send to network layer.

Network Layer

- ✓ At each step, packets final destination is compared with forwarding table details.
- ✓ Network layer gives back the packet to link layer.
- ✓ These two layers are only involved in intermediate steps so forming the core of routers.
- ✓ Routers can forward the packet in millisecond.
- ✓ Network layer recognizes when the packet reaches to its final destination.
- ✓ Packet is then handed over to transport layer which assembles the original message and hands over to application layer.

Port Numbers

- ✓ Determining which unit within the application layer should receive an incoming message is an important task of the transport layer. This is handled by assigning unique (**port numbers)**

Accepted Port Numbers

- ✓ Browser contacts HTTP server at port # 80. Similarly SMTP clients knows to contact application layers at server at port # 25

Summary

Layered Approach to Internet Software III

- ✓ Network Layer.
- ✓ Link Layer.
- ✓ Ports

Topic-080. TCP/IP protocol suite

Need

- ✓ Published standards so that manufacturers can supply equipment and software that could function well with products of other vendors.
- ✓ **Open System Interconnection (OSI)**

OSI Reference Model

- ✓ By International Organization for Standardization.
- ✓ Seven level hierarchy as compared to four level hierarchy we just learnt
- ✓ Remained slow in replacing the 4 level hierarchy as it was already in use.

TCP and IP

- ✓ TCP Suite is collection of large number of protocols to implement four level hierarchy
- ✓ Entire collection is referred to as TCP/IP-misleading.
- ✓ Transmission Control Protocol
- ✓ Internet Protocol

TCP version

- ✓ TCP defines a version of transport layer.
- ✓ More than one way of implementing transport layer either for TCP or **UDP (User Datagram protocol)**
- ✓ This is similar to choosing different shipping company

Differences between TCP and UDP

- ✓ TCP establishes a connection first while UDP does not.
- ✓ UDP is called connectionless protocol
- ✓ Destination might not be operational!

Differences between TCP and UDP

- ✓ TCP transport layers at origin and destination works together by acknowledgement and retransmission i.e. called reliable protocol.
- ✓ UDP unreliable protocol
- ✓ TCP can adjust the transmission rate to avoid control **Congestion.**

UDP benefits

- ✓ Transport layer based on UDP is more streamlined than TCP.
- ✓ UDP is perfect choice when receiver is ready like DNS Lookups, VOIP,
- ✓ Although in emails transfer, TCP is better choice.

IP

- ✓ Internet Protocol is a standard to implement the network layer (we have already talked).
- ✓ Forwarding and routing.
- ✓ Hop count: adding a value at origin by which time the packet should be forwarded normally 64 is sufficient.

Summary

TCP/IP Protocol Suite

- ✓ OSI
- ✓ TCP version
- ✓ UDP
- ✓ Differences between TCP and UDP
- ✓ IP

Topic-081. Security (forms of attacks)

Attacks

- ✓ Computer Systems and its contents can be attacked by many ways
- ✓ Many of these incorporates the Malicious Software (Malware)

Malicious Software Types

- ✓ Viruses
- ✓ Worms
- ✓ Trojan Horses
- ✓ Spyware

Virus

- ✓ A virus is software that infects a computer by inserting itself into program that already reside in the machine.
- ✓ When host is executed, virus also executed.
- ✓ Degrading portions of OS, erasing large blocks of mass storage, corrupting data and other programs

Worms

A worm is an autonomous program that transfers itself through a network, taking up residence in computers and forwarding copies of itself to other computers.

Trojan Horse

A Trojan horse is a program that enters a computer system disguised as a desirable program, such as a game or useful utility package, that is willingly imported by the victim.

Starts immediate or wait for some event,

Normally transferred using email attachments.

Spyware

- ✓ Sniffing Software.

collects information about activities at the computer

on which it resides and reports that information back to the instigator of the attack.

- ✓ Companies use to make customer profile.
- ✓ Searching passwords and credit card numbers

Phishing

- ✓ Unlike spyware asking explicitly the required information normally using an email from so-called Government, law-enforcement agency, financial institute, etc

Denial of Service (DOS) attacks

- ✓ process of overloading a computer with messages
- ✓ an attacker usually plants software on numerous unsuspecting computers (botnet) that will generate messages when a signal is given.

Spam

- ✓ Unwanted Junk emails.
- ✓ Medium for phishing and Trojan horses.

Summary

Security (Forms of Attacks)

- ✓ Viruses
- ✓ Worms
- ✓ Trojan Horses
- ✓ Spyware
- ✓ Phishing
- ✓ DOS
- ✓ SPAM

Topic-082. Protection & cures

What can be done!

- ✓ Prevention is better than cure!
- ✓ Firewall
- ✓ Spam Filters
- ✓ Proxy Servers
- ✓ Auditing Software
- ✓ Antivirus Software

Firewall

- ✓ Installed at the gateway of organization's intranet.
- ✓ Blocking messages coming from unknown regions (terminating DOS attacks)
- ✓ Blocking incoming messages having origin address of the region – Spoofing.
- ✓ Can protect individual computer rather than network.
- ✓ If a computer is not a server, then all access to server applications and ports should be blocked. This is one of the way to enter into the system.

Firewall – Spam Filters

- ✓ Specific Firewalls to block unwanted emails.
- ✓ Sophisticated technique to distinguish b/w desirable and spam emails.

Proxy Servers

- ✓ Acts as intermediary between a client and a server with the goal of shielding the client from adverse actions of the server.
- ✓ Server does not have an opportunity to learn about clients, their internal intranet structures.

- ✓ Can establish a proxy server for a particular kind of service (FTP, HTTP, Telnet etc).
- ✓ Another advantage is, like: FTP proxy server could check all incoming files for the presence of known viruses and block all infected files.

Auditing Software

- ✓ Deduction of sudden increase in message traffic
- ✓ Monitor activities of systems firewalls.
- ✓ Detection of irregularities

Antivirus Software

- ✓ used to detect and remove the presence of known viruses and other infections.
- ✓ Still not a guarantee for safety.
- ✓ User should not open emails from unknown sources.
- ✓ Do not respond to pop-up adds.
- ✓ Remove computer from internet when you do not need it

Summary

Protection and Cures

- ✓ Firewall
- ✓ Spam Filters
- ✓ Proxy Servers
- ✓ Auditing Software
- ✓ Antivirus Software

Topic-083. Encryption

Why?

- ✓ We learnt about how to secure the computer system
- ✓ But one of the main thread is to gain access to the data.
- ✓ One solution is password.
- ✓ It is not viable when the data is transferred.

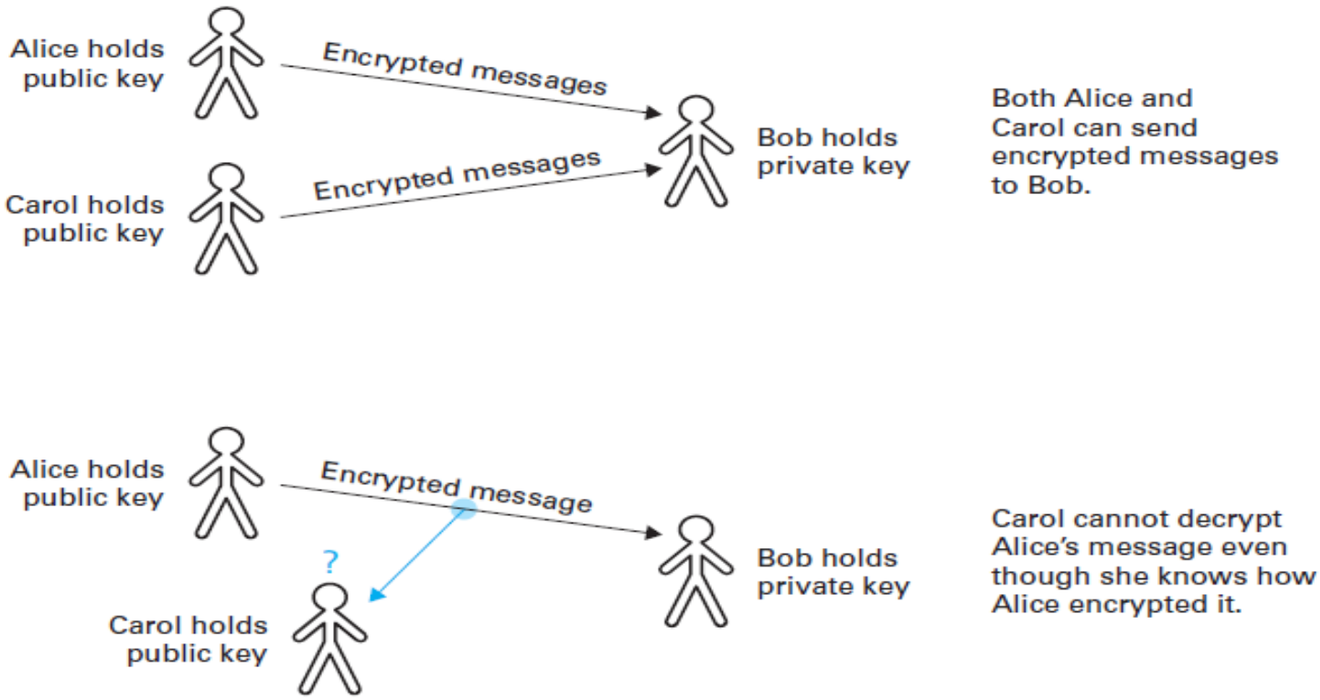
Secure version of HTTP

- ✓ Backbone of HTTPS is **Secure Sockets Layer (SSL)**
- ✓ Developed by Netscape to provide secure transmission b/w client and server.

Public Key Encryption

- ✓ Fascinating strategy.
- ✓ Having knowledge of how messages were encrypted does not allow other to decrypt messages.
- ✓ Use public and private keys.

Public Key Encryption – An Example



Problem might occur in such Encryption

- ✓ Making sure the public key is accurate for the destination system like using a wrong public key make your message waste for the destination and useful for hacker.

Certificate Authorities

- ✓ One of the solution to this problem.
- ✓ Servers provide reliable public-key information to their clients in packages known as **certificates**
- ✓ A certificate is a package containing a party's name and that party's public key.

Digital Signatures

- ✓ In some of the public key encryption, the role of encryption and decryption can be reversed.
- ✓ Encryption with the private key.
- ✓ The holder of the private key can produce a bit pattern, called a **digital signature,**
- ✓ The message is revealed to be authentic.

Summary

Encryption

- ✓ HTTPs
- ✓ Public and private keys
- ✓ Certificate Authorities
- ✓ Digital Signatures

Topic-084. Legal approaches to Network security

Yet another solution!

- ✓ Enhancing the security of computer networking systems is to apply legal remedies.
- ✓ Defining what is legal and what is illegal in different contexts.
- ✓ Illegal actions may lead to punishment.

Two Issues

- ✓ Making an action illegal does not stop the action.
- ✓ International law, Illegal in one country and legal in another country.

US – Computer Fraud and Abuse Act

- ✓ Passed in 1984 and has been revised many times.

- ✓ Covers:
  - ✓ Worms and Viruses
  - ✓ Theft of Information
  - ✓ Many more related things are covered anything of value.

US – Electronic Communication Privacy Act (ECPA)

- ✓ Employers/ISP rights to look into the communications of employees.
- ✓ Illegal for ISP to make available the private information
- ✓ Employers can access information especially when the equipment of employer is used?

Pakistan – CyberSecurity Law

- ✓ National Assembly Passes New Cybercrime Law. In 2016.
- ✓ Punishment of 10 years for first offenders and 20 years for repeat offenders
- ✓ http://thehill.com/policy/cybersecurity/265285-judges-struggle-with-cyber-crime-punishment
- ✓ Covers many security concerns e.g.,
  - ✓ Hacking
  - ✓ Phishing
  - ✓ Malware
  - ✓ Hacking tools
  - ✓ Identity Theft/fraud.
  - ✓ Electronic theft.

Pakistan – Applicable security Acts

- ✓ Prevention of Electronic Crimes Ordinance, 2007
- ✓ Electronic Transaction Ordinance, 2002, 2008
- ✓ Pakistan Telecommunicaiton (Re-organization) Act, 1996
- ✓ Wireless telegraphy Act, 1933
- ✓ Telegraph Act, 1885.
- ✓ Federal Investigation Agency Act, 1974
- ✓ Payments and Electronic Fund Transfers Act, 2007

Summary

Legal to Network Security

- ✓ Possibilities/Issues
- ✓ US acts
- ✓ Pakistan Cyber Security Law
- ✓ Applicable other laws

Topic-085. An informal review

Algorithm

- ✓ Set of steps that define how a task is performed.
- ✓ Lets start discussing simple Algorithms

Making a Cup of Tea

1. Put water in kettle
2. Boil water
3. Put Tea in water
4. Add Milk
5. Stir
6. Add Sugar
7. Put tea in cup
8. Fetch cup

Convert KM to Meters

1. Take input of KM's
2. Multiply the input with 1000.
3. Display the result

Machine Cycle

As long as the halt instruction has not been executed continue to execute the following steps:

a. Fetch an instruction.

b. Decode the instruction.

c. Execute the instruction.

Running Example

Mean Algorithm

1. List of numbers is {1, 2, 5, 6, 6, 99}
2. Set a total to 0
3. Set a counter to 0
4. Start with the first number in the list
    a. Add the number to the total
    b. Add one to the counter
    c. Is this the last number in the list?
        i. If no, go to the next number and repeat steps a – c
        ii. If yes, go to step 5
5. Find the mean by dividing the total by the value of the counter
6. Display the mean

Algorithm Execution

Many researchers believe that every activity of the human mind, including imagination, creativity, and decision making, is actually the result of algorithm execution.

Summary

Algorithm – An Informal Review

- ✓ Informal definition
- ✓ Examples

Topic-086. Formal definition of algorithm

Formal Definition

An algorithm is an ordered set of unambiguous, executable steps that defines a terminating process.

Order vs sequence

- ✓ Have a well-established structure in terms of the order of their execution
- ✓ Does not mean a sequence – like we do not have in parallel processing.
- ✓ Flip-flops producing their output individually and then together they have a meaning.

Executable

- ✓ Make a list of all the positive integers.
- ✓ Computer scientists use the term effective to capture the concept of being executable.

Unambiguous

The information in the state of the process must be sufficient to determine

uniquely and completely the actions required by each step.
Make a pretty cartoon!

Terminating Process

The execution of an algorithm must lead to an end.
But!
Monitoring the vital signs of a hospital patient Maintaining an aircraft's altitude in flight

Summary

Formal Definition of Algorithm

- ✓ Order vs Sequence
- ✓ Executable
- ✓ Unambiguous
- ✓ Terminating Process

Topic-087. Abstract nature of algorithms

Algorithm and its Representation

- ✓ Algorithm is abstract, its representation may change.
- ✓ Story and a book
- ✓ Story is abstract, physical book is one representation.
- ✓ Translation to another language is another representation.

Celsius to Fahrenheit

- ✓ F = (9/5)C + 32

  OR
- ✓ Multiply the temperature reading in Celsius by 9⁄5 and then add 32 to the product

  OR
- ✓ Form of an electronic circuit.

Sometime Confusing

- ✓ Meteorologists understands:

"Convert the Celsius reading to its Fahrenheit equivalent"

- ✓ However, layperson can argue its ambiguous
- ✓ concept of primitives will solve this in next module.

Program and Process

- ✓ Formal representation of an algorithm is program while the activity of executing a program is called process – executing an algorithm

Summary

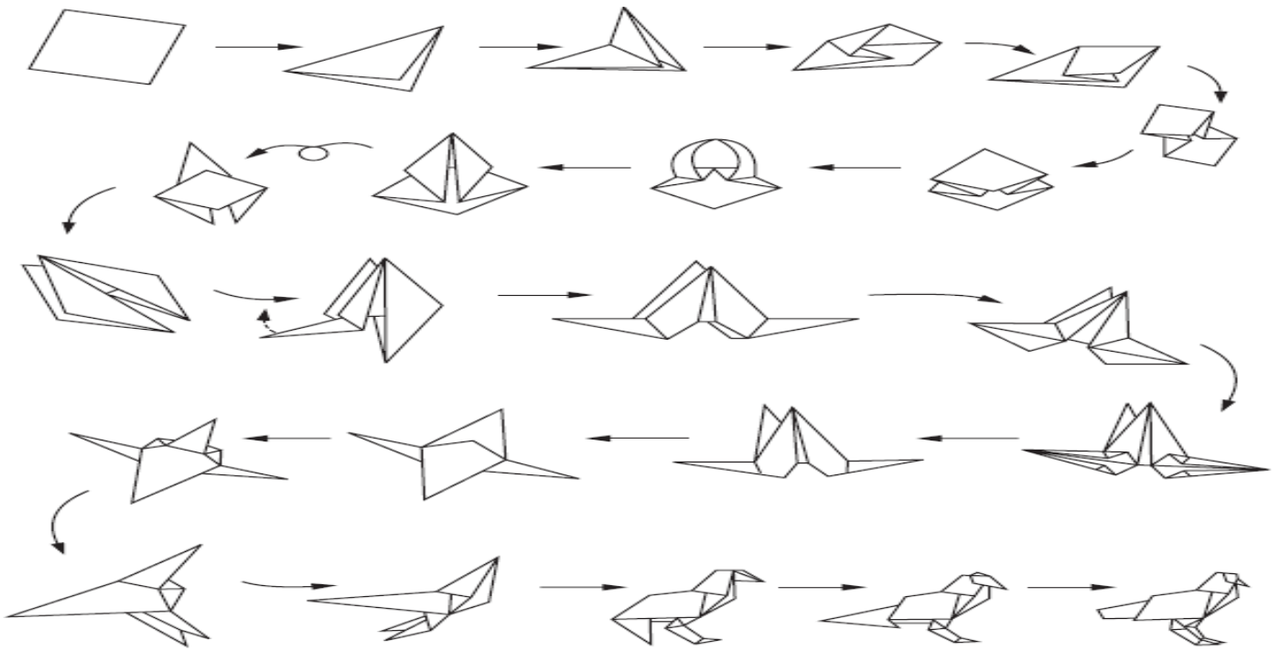
Abstract Nature of Algorithm

- ✓ Algorithm and its representation
- ✓ Example
- ✓ Confusions
- ✓ Program and process

Topic-088. Representation (Primitives)

Algorithm Representation

- ✓ Representation of an algorithm requires some language.
- ✓ In case of human this could Urdu, English, German etc. or language of pictures
- ✓ Sometime misunderstanding!
- ✓ Visiting Grandchildren can be problematic! Have two meaning.



- ✓ Student of origami would have little difficult to follow the instructions as compared to other students.

Primitive

- ✓ Computer Science establishes set of building blocks from which algorithm can be constructed, such building blocks are known as primitive.

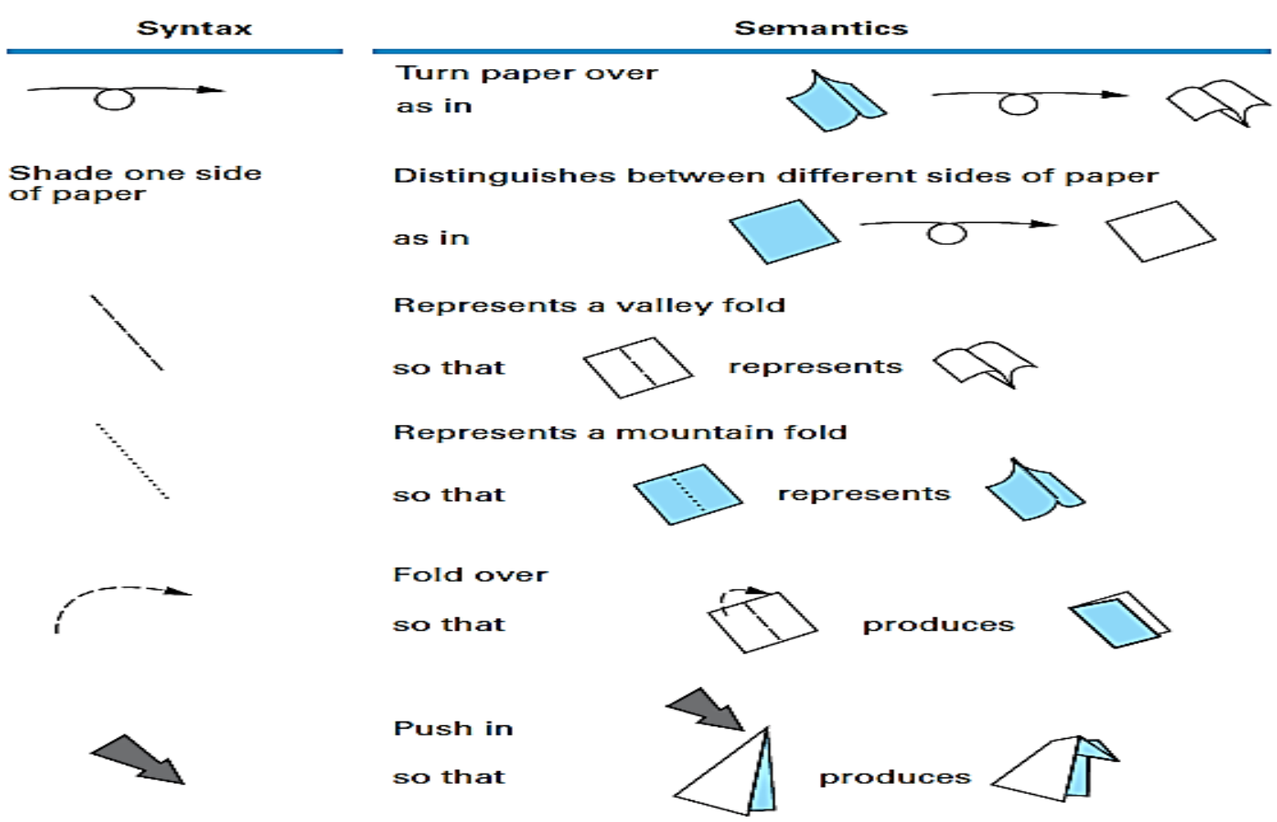
Programming Language

- ✓ A collection of primitives along with a collection of rules stating how the primitives can be combined to represent more complex ideas constitutes a programming language.

Primitives

- ✓ Each primitive has its own syntax (symbol) and semantics (meaning)

Origami Primitives



Representation in machine level

- ✓ Machine level representation is ultimately required.

- ✓ It is tedious
- ✓ Therefore, high level programming language have been built, mapping primitives to machine level instructions

Summary

Representation (Primitives)

- ✓ Primitives
- ✓ Origami
- ✓ Programming language
- ✓ Machine representation

Topic-089. Representation (Pseudocode)

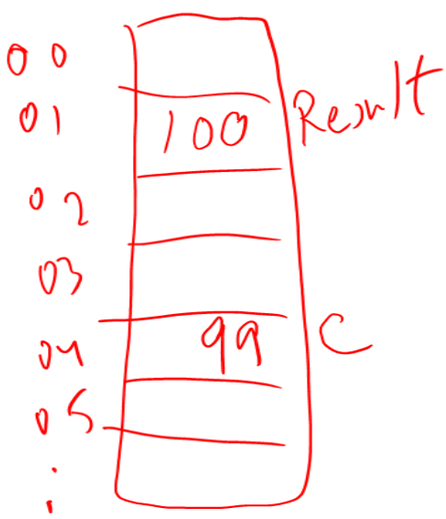
Definition

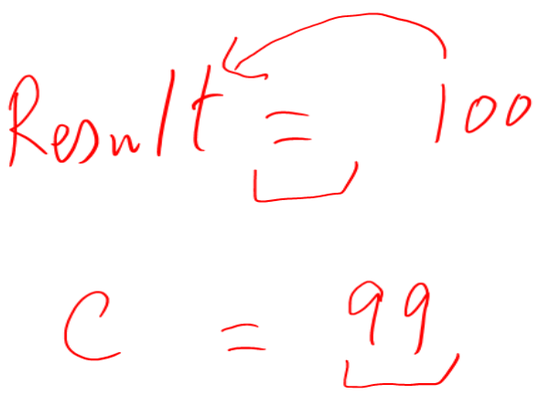
- ✓ Notational system in which ideas can be expressed informally during the algorithm development process

Writing Pseudocode

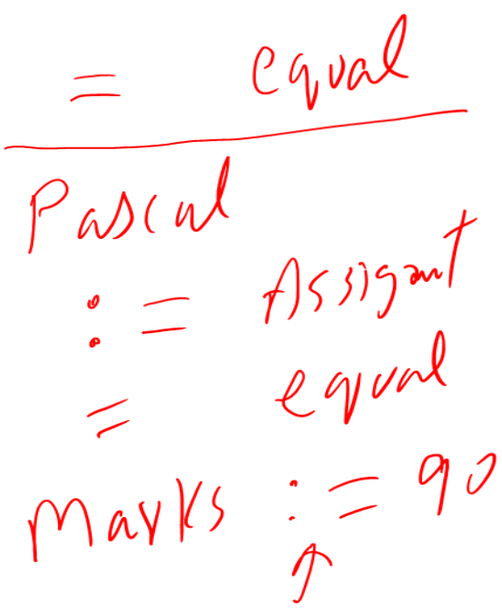
- ✓ One way is:
- ✓ Loosen the rules of formal programming language.
- ✓ Borrowing the syntax-semantics structure of the language.
- ✓ Result is: many variants exist depending on what language is followed.

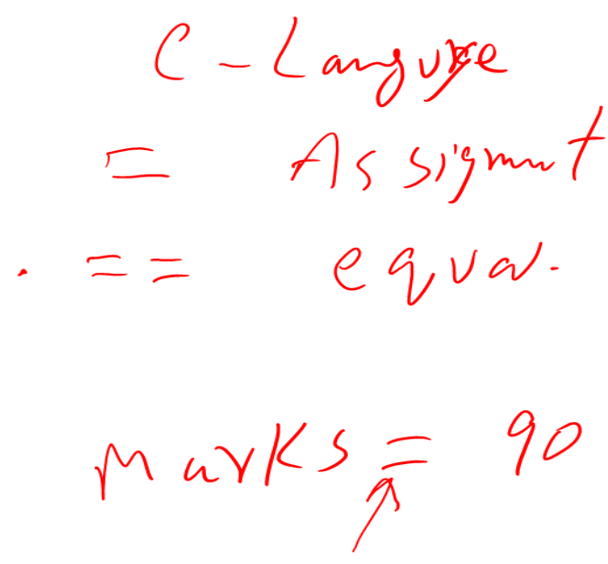
Concept of Saving Results – Marks example.





Assignment and equal operator in Pascal and C.





Pseudocode Language selection

- ✓ Popular are Algol and Pascal languages.
- ✓ Recently, Java and C based pseudocode are being used.
- ✓ Pseudocode must be consistent.
- ✓ We will use Python-like syntax in next modules.

If-structure:

```
If (marks >= 50):
    declare Pass
else
    declar Fail
```

Summary

Representation(Pseudocode)

- ✓ Definition
- ✓ Writing Pseudocode
- ✓ Assignment and equal structure
- ✓ If-structure

Topic-090. Representation (Pseudocode) While-Structure

Definition

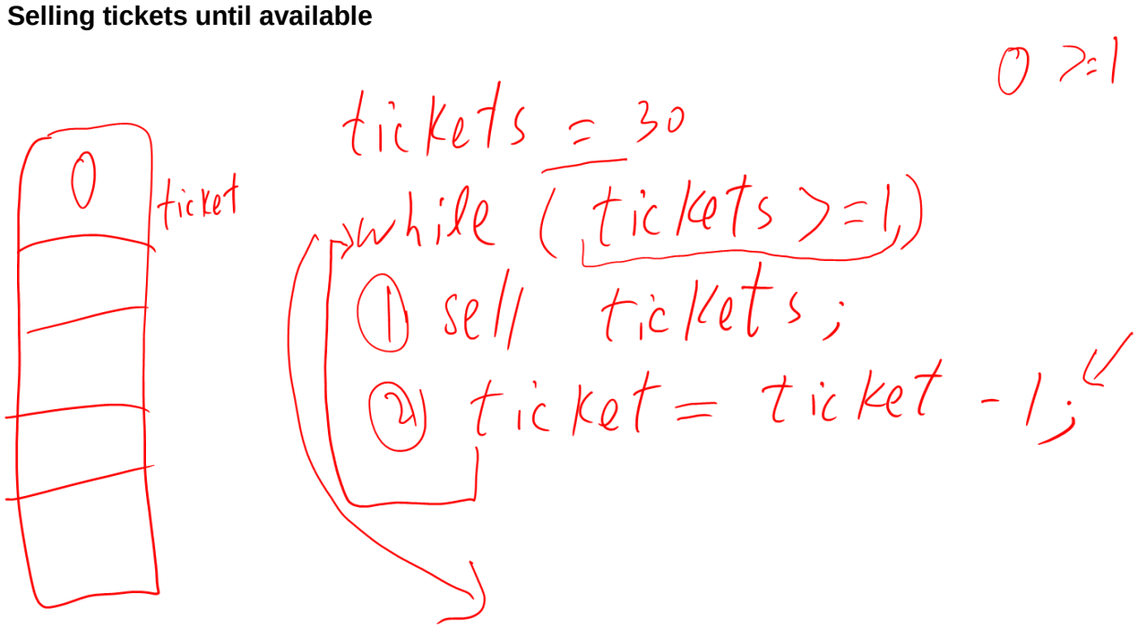
- ✓ Repeated execution of a statement or sequence of statements as long as some condition remains true.

Example

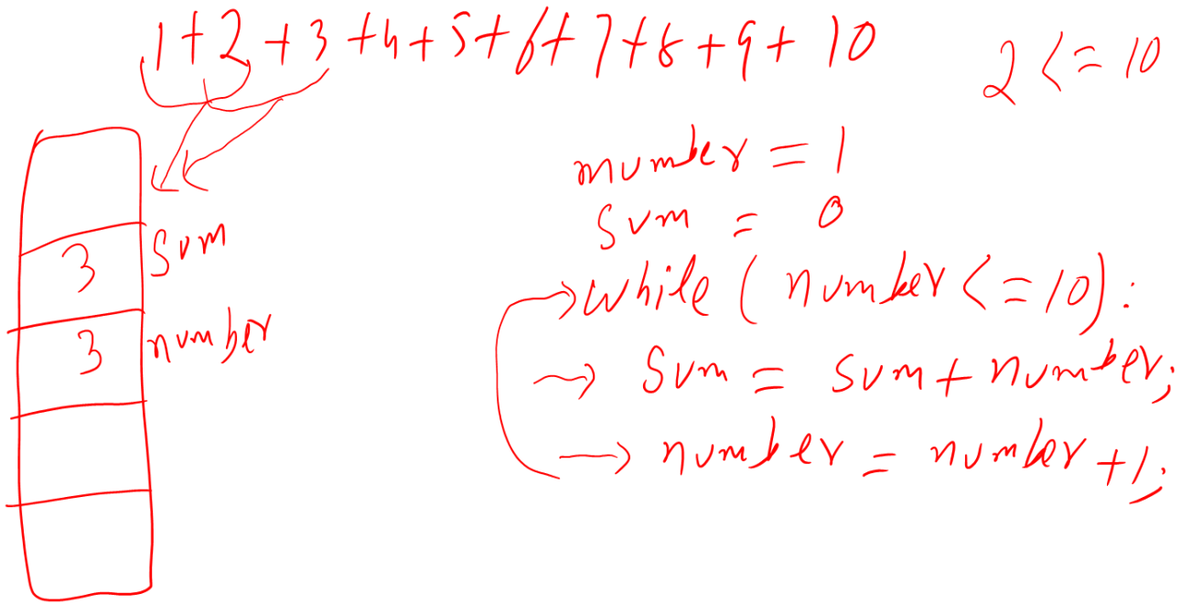
- ✓ As long as there are tickets to sell, continue selling tickets.
- ✓ While there are tickets to sell, keep selling tickets.

while (tickets remain to be sold):

sell a ticket
Selling tickets until available



Calculating Sum of numbers ranging from 1-10



Summary

Representation(Pseudocode)

- ✓ While structure with examples

Topic-091. Representation (Pseudocode) Function-Structure

Why we need

- ✓ We want to describe activities that can be used as abstract tools in other applications.
- ✓ Different names:
  - ✓ Subprogram
  - ✓ Subroutine
  - ✓ Procedure
  - ✓ Method
  - ✓ **Function**

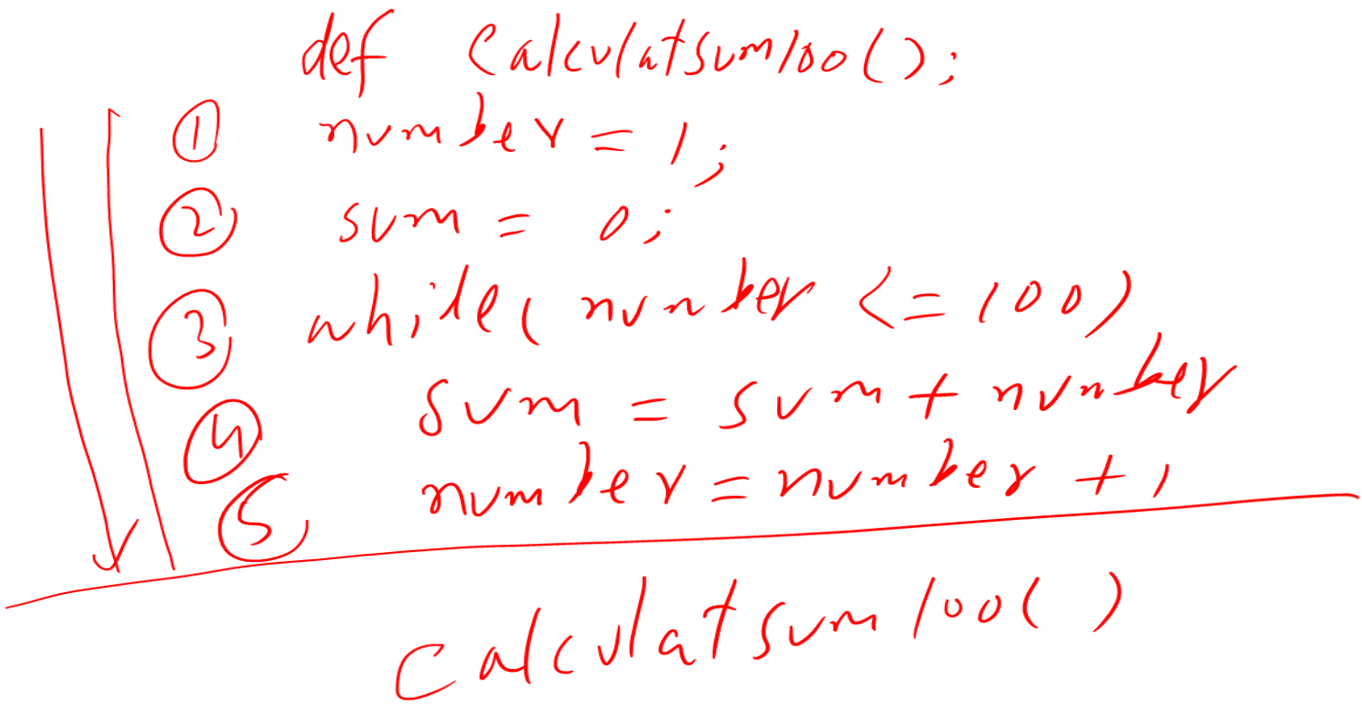
How to use Functions

- ✓ def name():
- ✓ Parameters:
- ✓ def Sort (List):

Example

```
def Greetings():
Count = 3
while (Count > 0):
print('Hello')
Count = Count – 1
```

Example – Calculating sum of numbers 1-100



Example

```
if (not raining):
  if (temperature == hot):
    go swimming
  else:
    play golf

else:
  watch television
```

Summary

Representation(Pseudocode)

- ✓ Function Structure
- ✓ Indentation
- ✓ Examples

Topic-092. Discovery (The Art of Problem Solving)

Stages

Phase 1. Understand the problem.

Phase 2. Get an idea of how an algorithmic function might solve the problem.

Phase 3. Formulate the algorithm and represent it as a program.

Phase 4. Evaluate the program for accuracy and for its potential as a tool for solving other problems.

Important points

- ✓ You do not solve by following rather taking lead and initiative.
- ✓ Step by step procedure does not work,
- ✓ Get involved, you will notice, you have solved using the phases.

Problem Example

Person A is charged with the task of determining the ages of person B's three children. B tells A that the product of the children's ages is 36. After considering this clue, A replies that another clue is required, so B tells A the sum of the children's ages. Again, A replies that another clue is needed, so B tells A that the oldest child plays the piano. After hearing this clue, A tells B the ages of the three children.
How old are the three children?.

Solution Step (a)

a. Triples whose product is 36

(1,1,36)	(1,6,6)
(1,2,18)	(2,2,9)
(1,3,12)	(2,3,6)
(1,4,9)	(3,3,4)

Solution Step (b)

a. Triples whose product is 36		**b.** Sums of triples from part (a)	
(1,1,36)	(1,6,6)	1 + 1 + 36 = 38	1 + 6 + 6 = 13
(1,2,18)	(2,2,9)	1 + 2 + 18 = 21	2 + 2 + 9 = 13
(1,3,12)	(2,3,6)	1 + 3 + 12 = 16	2 + 3 + 6 = 11
(1,4,9)	(3,3,4)	1 + 4 + 9 = 14	3 + 3 + 4 = 10

Summary

Algorithm: Discovery (the art of solving the problem)

- ✓ Phases
- ✓ Example

Topic-093 Getting your foot in the door

Strategies

- ✓ There are many strategies to be used in different contexts.
- ✓ One important underlying point is "Get your foot in the door".
- ✓ Lets have an example

Example

Before A, B, C, and D ran a race they made the following predictions:

A predicted that B would win.

B predicted that D would be last.

C predicted that A would be third.

D predicted that A's prediction would be correct.

Only one of these predictions was true, and this was the prediction made by the winner. In what order did A, B, C, and D finish the race?

Example

Before A, B, C, and D ran a race they made the following predictions:

A predicted that B would win.

B predicted that D would be last.

C predicted that A would be third.

D predicted that A's prediction would be correct.

Only one of these predictions was true, and this was the prediction made by the winner. In what order did A, B, C, and D finish the race?

Example

Before A, B, C, and D ran a race they made the following predictions:

A predicted that B would win.

B predicted that D would be last.

C predicted that A would be third.

D predicted that A's prediction would be correct.

Only one of these predictions was true, and this was the prediction made by the winner. In what order did A, B, C, and D finish the race?

Example

Before A, B, C, and D ran a race they made the following predictions:

A predicted that B would win.

B predicted that D would be last.

C predicted that A would be third.

D predicted that A's prediction would be correct.

Only one of these predictions was true, and this was the prediction made by the winner. In what order did A, B, C, and D finish the race?

Example

Before A, B, C, and D ran a race they made the following predictions:

A predicted that B would win.

B predicted that D would be last.

C predicted that A would be third.

D predicted that A's prediction would be correct.

Only one of these predictions was true, and this was the prediction made by the winner. In what order did A, B, C, and D finish the race?

What it requires!

- ✓ Not blindly following, it requires the creativity!

Summary

Algorithm: Discovery (Get You Foot in the Door)

- ✓ How to approach the problem
- ✓ Example

Topic-094. Algorithm discovery strategies-1

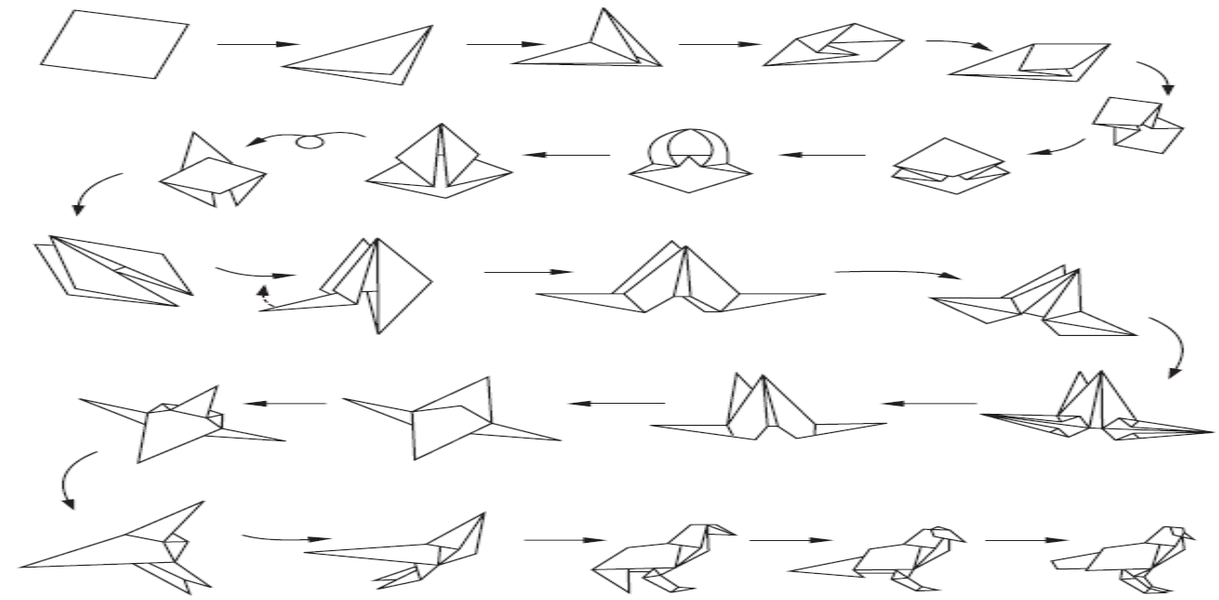
Strategies

- ✓ Reverse Engineering
- ✓ Look for Related Problems

Reverse Engineering

- ✓ When the problem is to find a way of producing a particular output from a given input.
- ✓ Example bird folding algorithm

Reverse Engineering-Example



Look for Related Problems

- ✓ This technique is of particular value in the context of program development.
- ✓ Program development is not the process of solving a particular instance of a problem but rather of finding a general algorithm

Sorting the Names

- ✓ Abida, Omer, Ahmad, Ayesha
- ✓ Khalida, Rehman, Awais, Waqas

Look for Related Problems

- ✓ Finding/developing an algorithm that could work in all situations

Summary

Algorithm: Discovery (Strategies-I)

- ✓ Reverse Engineering
- ✓ Look for Related Problem

Topic-095. Algorithm discovery strategies-2

Strategies

- ✓ Stepwise Refinement/Top-down Methodology
- ✓ Bottom-up Methodology

Stepwise Refinement/Top-down Methodology

- ✓ Breaking up original problem into sub problems until the entire problem has been reduced to a collection of easily solved sub-problems.

Stepwise Refinement/Top-down Methodology - Example

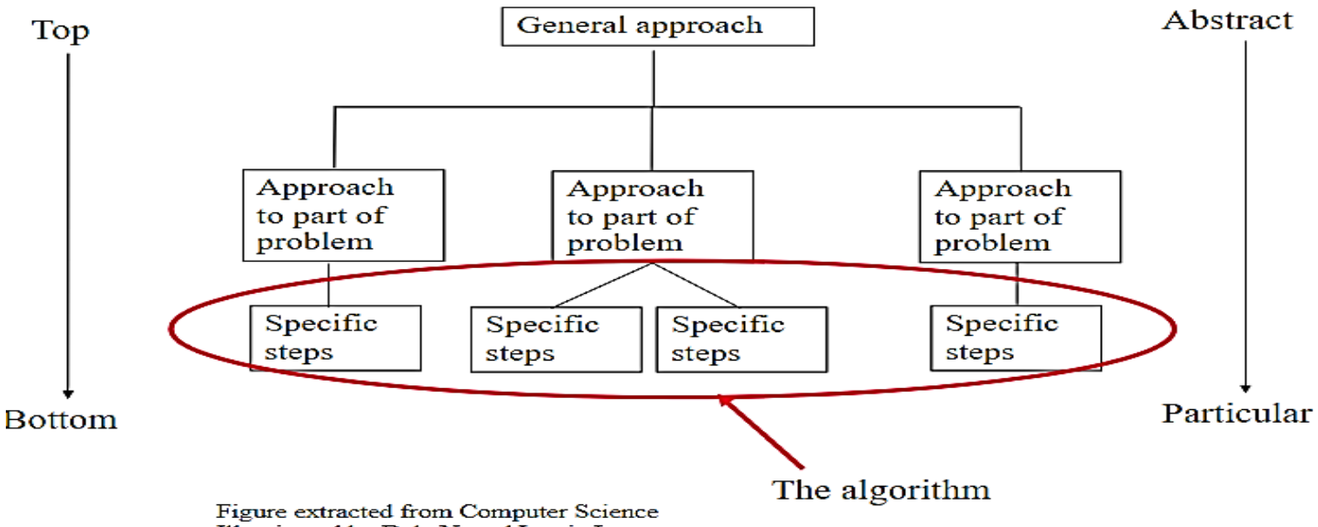
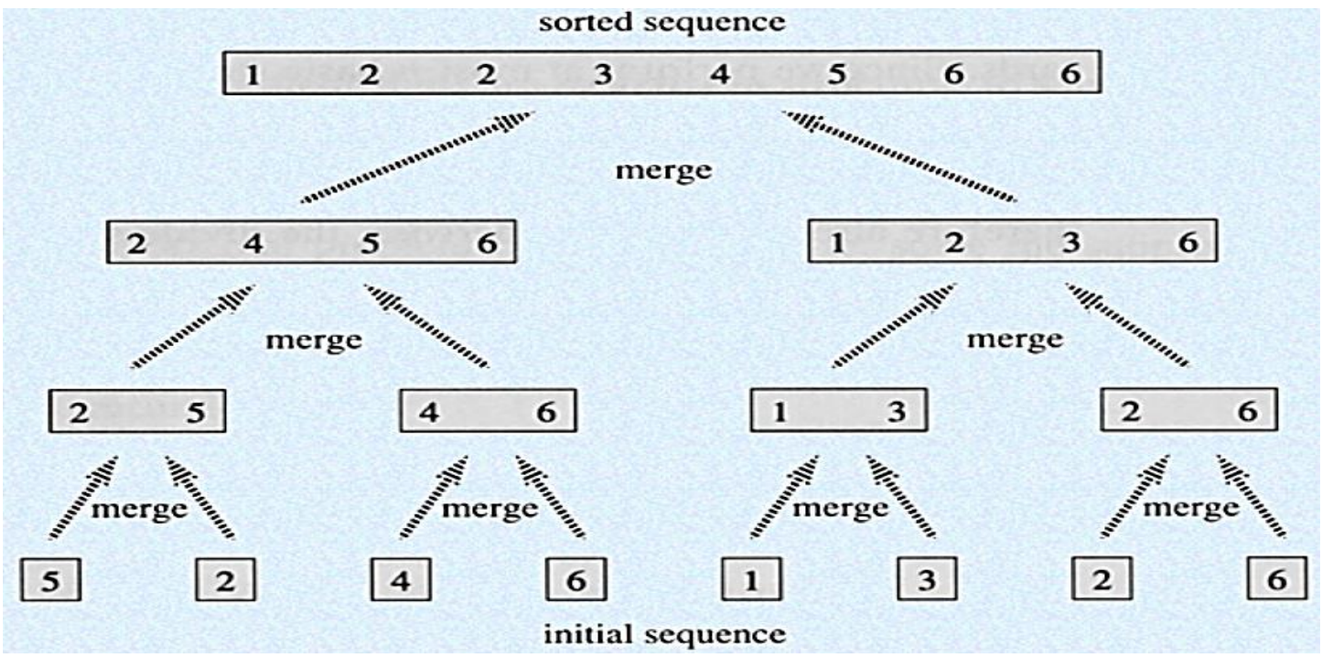


Figure extracted from Computer Science Illuminated by Dale N. and Lewis J.

Bottom-up Methodology

- ✓ progresses from the specific to the general.
- ✓ Example

Stepwise Refinement/Bottom-up Methodology - Example



Top-down & Bottom-up

- ✓ Both approaches complement each other.
- ✓ The decomposition of a problem proposed by the top-down methodology of stepwise refinement is often guided by the problem solver's intuition, which might be working in a bottom-up mode.

Summary

Algorithm: Discovery (Strategies-II)

- ✓ Stepwise Refinement/Top-down Methodology
- ✓ Bottom-up Methodology

Topic-096. Iterative Structures (Sequential Search Algorithm)

Sequential Search

- ✓ Searching from the start to the end of the list until we find the required value or list ends
- ✓ Assumption: List is sorted.

Sequential Search

12	18	29	38	45	89	101	500	801	999

Select the first entry in the list as TestEntry.

while (TargetValue > TestEntry and entries remain):

Select the next entry in the list as TestEntry.

Select the first entry in the list as TestEntry.

while (TargetValue > TestEntry and entries remain):

Select the next entry in the list as TestEntry.

if (TargetValue == TestEntry):

Declare the search a success.

else:

Declare the search a failure.

Sequential Search Function

```
def Search(List, TargetValue):
  if (List is empty):
    Declare search a failure.
  else:
    Select the first entry in List to be TestEntry.
    while (TargetValue > TestEntry and
         there remain entries to be considered):
      Select the next entry in List as TestEntry.
    if (TargetValue == TestEntry):
      Declare search a success.
    else:
      Declare search a failure.
```

Summary

Algorithm: Iterative Structures (Sequential Search Algorithm)

- ✓ Developed an algorithm from scratch and represented it in pseudocode

Topic-097. Iterative structures (Loop Control)

Loops

- ✓ The repetitive use of an instruction or sequence of instructions is an important algorithmic concept.
- ✓ One way is using loop.

Execution Steps

check the condition.

execute the body.

check the condition.

execute the body.

check the condition.

until the condition fails

Example – Writing Hello 5 times

Example – printing counting from 1 to 20

print (1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20)

Summary

Algorithm: Iterative Structures (Loop Control)

- ✓ Loop execution with examples

Topic-098. Iterative Structures (Components of Repetitive Control)

Initialize: Establish an initial state that will be modified toward the termination condition

Test: Compare the current state to the termination condition and terminate the repetition if equal

Modify: Change the state in such a way that it moves toward the termination condition

Examples of Finite loops

Examples of Infinite loops

Summary

Algorithm: Iterative Structures (Components of Repetitive Control)

- ✓ Components of repetition
- ✓ Finite Loops
- ✓ Infinite Loops

Topic-099. Iterative Structures: Loop Execution (Example-1)

Finding Maximum Number

12	18	299	38	999	89	101	500	801	45

Summary

Algorithm: Iterative Structures (Loop Execution Example-1)

- ✓ Developed the algorithm for finding maximum number in a list

Topic-100. Iterative Structures: Loop Execution (Example-II)

Finding Factorial of a Number ‘n’

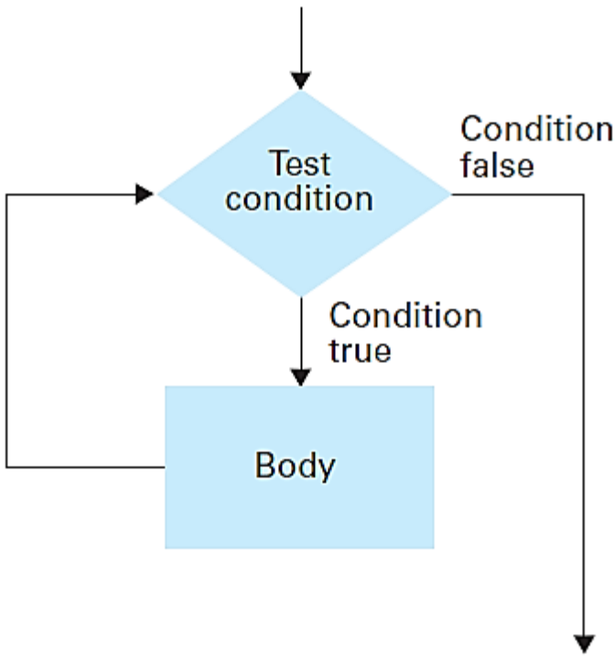
Summary

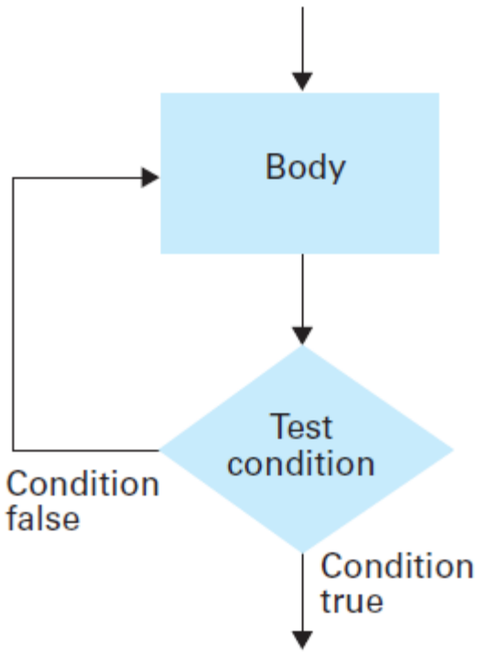
Algorithm: Iterative Structures (Loop Execution Example-1I)

- ✓ Developed the algorithm for finding factorial of n

101. Iterative Structures: (Pretest and Posttest loops)

Pretest and Posttest Loops





Pseudocode

- ✓ Python does not have such loop, so we are borrowing from other language to represent it

Posttest Loop representation

repeat:

take a coin from your pocket

until (there are no coins in your pocket)

For-each loop

Sum = 0

for Number in List:

Sum = Sum + Number

For-each loop example

(Finding number of Pass students)

Summary

Algorithm: Iterative Structures (Pretest and Posttest Loops)

- ✓ Flowchart
- ✓ Pseudocode
- ✓ For-each loop

102. Insertion Sort Algorithm

Sorting in Alphabetic Order

Fred

Alex

Diana

Byron

Carol

Constraints

- ✓ Sorting the list within itself.
- ✓ Moving to other list is not allowed.
- ✓ Utilizing space efficiently

Sorting Procedure

Sublist consisting of only the top name, Fred, is sorted but the sublist consisting of the top two names, Fred and Alex, is not

Sorting Procedure

Fred

Byron

Alex

Carol

Diana

Summary

Algorithm: Insertion Sort Algorithm

- ✓ Insertion Sort Procedure

103. Insertion sort algorithm example

Remember the problem from previous module

Fred

Byron

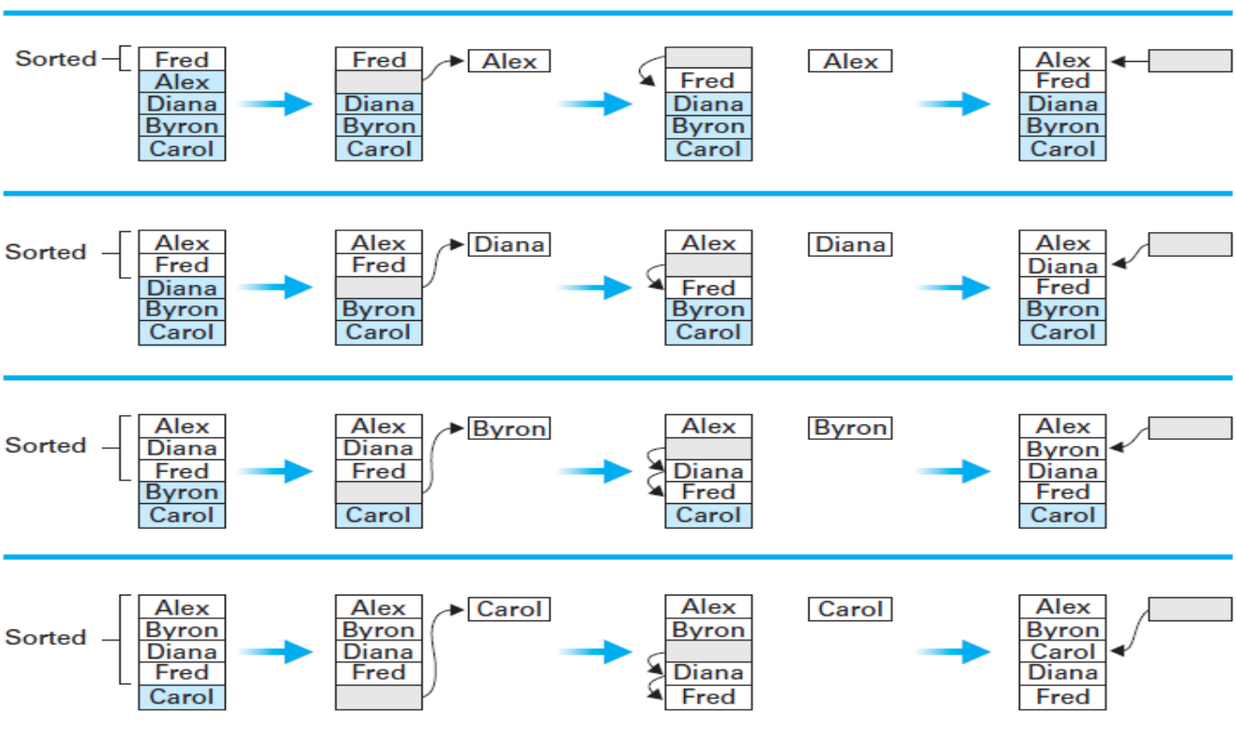
Alex

Carol

Diana

Sorting Procedure

Sublist consisting of only the top name, Fred, is sorted but the sublist consisting of the top two names, Fred and Alex, is not



Summary

Algorithm: Insertion Sort Algorithm example

- ✓ Insertion Sort Procedure
- ✓ Complete solved example

104. Recursive structure: the binary search algorithm

Recursion

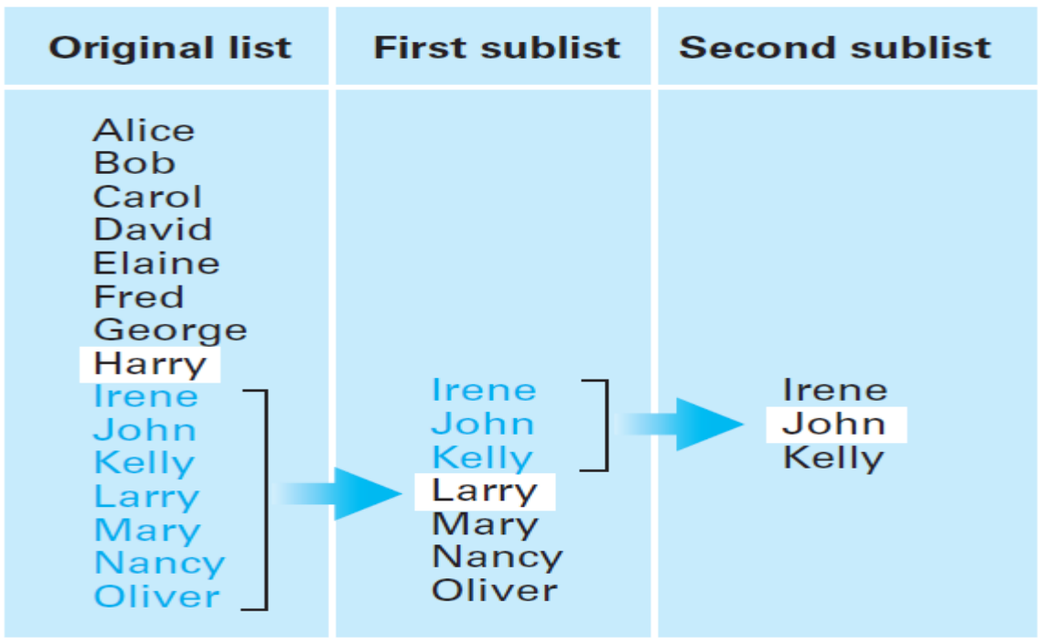
- ✓ Alternative way to loop paradigm.
- ✓ Loop executes set of instructions and repeats, however, recursive structure repeats as subtask.

- ✓ Telephone hold example

Binary Search Algorithm

- ✓ We have a sorted list and wants to search within the list.
- ✓ How to search a dictionary???
- ✓ Starting from the Middle.

Example



Algorithm – First Draft

```
if (List is empty):
  Report that the search failed.
else:
  TestEntry = the "middle" entry in the List
  if (TargetValue == TestEntry):
    Report that the search succeeded.
  if (TargetValue < TestEntry):
    Search() the portion of List preceding TestEntry for TargetValue,
      and report the result of that search.
  if (TargetValue > TestEntry):
    Search() the portion of List following TestEntry for TargetValue,
      and report the result of that search.
```

Algorithm – Pseudocode

```
def Search(List, TargetValue):
  if (List is empty):
    Report that the search failed.
  else:
    TestEntry = the "middle" entry in List
    if (TargetValue == TestEntry):
      Report that the search succeeded.
    if (TargetValue < TestEntry):
      Sublist = portion of List preceding
        TestEntry
      Search(Sublist, TargetValue)
    if (TargetValue > TestEntry):
      Sublist = portion of List following
        TestEntry
      Search(Sublist, TargetValue)
```

Summary

Algorithm: Recursive (Binary Search Algorithm)

- ✓ Recursion
- ✓ Binary Search Algorithm
- ✓ Example
- ✓ Pseudocode

105. Recursive control

Binary Search Algorithm

```
def Search(List, TargetValue):
  if (List is empty):
    Report that the search failed.
  else:
    TestEntry = the "middle" entry in List
    if (TargetValue == TestEntry):
      Report that the search succeeded.
    if (TargetValue < TestEntry):
      Sublist = portion of List preceding
        TestEntry
      Search(Sublist, TargetValue)
    if (TargetValue > TestEntry):
      Sublist = portion of List following
        TestEntry
      Search(Sublist, TargetValue)
```

Example search Zahid

1. Abida
2. Ahmad
3. Asif
4. Ayesha
5. Afzal
6. Bashir
7. Fawad
8. Khalid
9. Omer
10. Rahat
11. Usama
12. Yahyah
13. Younas
14. Zahid
15. Zubair

Example search Ajmal

1. Abida
2. Ahmad
3. Asif
4. Ayesha
5. Afzal
6. Bashir
7. Fawad
8. Khalid
9. Omer
10. Rahat
11. Usama
12. Yahyah
13. Younas
14. Zahid
15. Zubair

Binary Search Algorithm

- ✓ Base Case
- ✓ Degenerative Case

```
def Search(List, TargetValue):
  if (List is empty):
    Report that the search failed.
  else:
    TestEntry = the "middle" entry in List
    if (TargetValue == TestEntry):
      Report that the search succeeded.
    if (TargetValue < TestEntry):
      Sublist = portion of List preceding
        TestEntry
      Search(Sublist, TargetValue)
    if (TargetValue > TestEntry):
      Sublist = portion of List following
        TestEntry
      Search(Sublist, TargetValue)
```

Summary

Algorithm: Recursive Control

- ✓ Recursion examples
- ✓ How to stop the recursion
- ✓ Base case/degenerative case

106. Algorithm efficiency

What is Algorithm Efficiency?

- ✓ A measure of the execution time necessary for an algorithm to complete work on a set of data.

Why need to measure efficiency?

- ✓ Although computer can execute billion of instructions in a second.
- ✓ However, decision between efficient and inefficient algorithm makes a great difference

Example

- ✓ University Registrar have 30, 000 students
- ✓ Sequential Search and Binary Search

Sequential Search Worst Case

- ✓ 30, 000 entries needs to be searched

- ✓ If retrieving and checking takes 10 millisecond.
- ✓ 30,000 * 10/ 1000

= 300 seconds = 5 mins

Binary Search (Worst case)

- ✓ 15 * 10/ 1000

= 0.15seconds

Algorithm Analysis

- ✓ Worst case
- ✓ Average case
- ✓ Best case

Sequential Search Average Case

- ✓ On average 15, 000 need to be searched
- ✓ If retrieving and checking takes 10 millisecond.
- ✓ 15,000 * 10/ 1000

= 150 seconds

=2.5mins

Summary

Algorithm: Efficiency

- ✓ What is Algorithm Efficiency
- ✓ Why we need to measure
- ✓ Example of sequential search and binary search

107. Software verification

What is Software Verification?

- ✓ Verification makes sure that the product is designed to deliver all functionality to the customer
- ✓ Checking software correctness .

Example

What is Software Verification?

- ✓ It answers the question:

“Am I building the software right?”

Started in the start of the development activity

Advantages?

- ✓ Decrease the count of defect in the later stage of development
- ✓ Helps to understand the product at the start.
- ✓ Reduce the chances of the failure.
- ✓ Helps in building the product as per the customer satisfaction.

Summary

Software Verification

- ✓ What is Software Verification
- ✓ Example
- ✓ Advantages.

108. Software verification example

Software Verification?

- ✓ Distinction between Program believed to be correct and the program that is correct.

Software Failure?

- ✓ Fatal Therac-25 X-ray Radiation
- ✓ In 1986, a man in Texas received between 16,500-25,000 radiations in less than 10 sec, over an area of about 1 cm.
- ✓ He passed away 5 months later.
- ✓ The root cause of the incident was a SW failure.

Welsch NHS IT Failure

- ✓ In 2018, doctors and staff were unable to access patients files.
- ✓ This was a technical issue of software not the security issue.

TSB bank customers

- ✓ In April 2018, million of customers experienced many issues due to moving to new banking platform:
  - ✓ Login issues
  - ✓ Customers were seeing details of other people account.
  - ✓ Inaccurate credits and debits

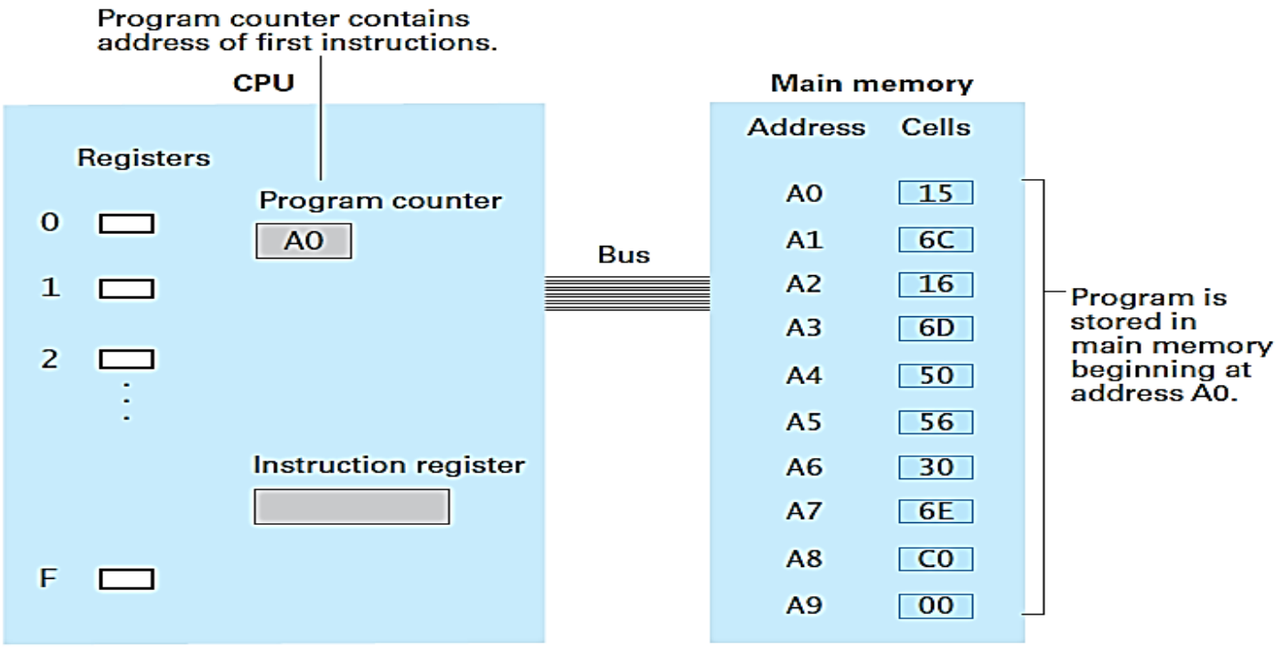
Summary

Software Verification Examples

- ✓ Learned with examples that the software verification is very important to avoid great loss in future!

Topic-109. Early generations-1

Do you remember this?



1. 156C
2. 166D
3. 5056
4. 306E
5. COOO

Using mnemonics and descriptive names

1. LD R5,Price
2. LD R6,ShippingCharge
3. ADDI R0,R5 R6
4. ST R0,TotalCost
5. HLT!

Assemblers

Converting mnemonics expressions into Machine language instructions

Assembly Language

A mnemonic system for representing is collectively called Assembly Language.

Program Variables or Identifiers

Such descriptive names are called program variables or identifiers

Summary

Programming Languages: Early Generations-I

- ✓ Machine Language
- ✓ Assemblers
- ✓ Assembly Language
- ✓ Concept of Variable

Topic-110. Early generations-2

Assembly Language-Disadvantages

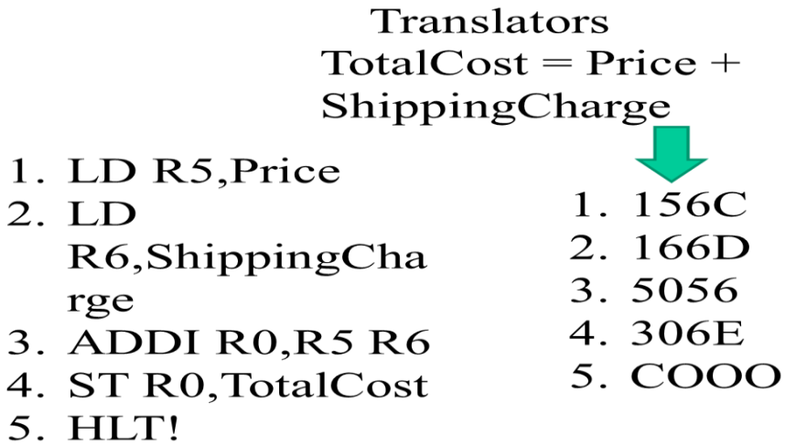
- ✓ Code is Machine dependent.
- ✓ Forced to think in small incremental steps.

Example:

TotalCost = Price + ShippingCharge

Machine Independent

- ✓ Third generation language, example FORTRAN-Formula Translator
- ✓ COBOL – Common Business Oriented Language



Compilers

- ✓ Such translation programs were called Compilers

Interpreters

- ✓ These programs were similar to translators except that they executed the instructions as they were translated instead of recording the translated version for future use.

Natural Languages and Formal Languages

- ✓ Computer languages follow strict rules and are precisely defined by grammars whereas natural languages evolved without formal grammatical analysis.

Summary

Programming Languages: Early Generations-II

- ✓ Assembly Language disadvantages
- ✓ Machine independence
- ✓ Translators
- ✓ Compilers
- ✓ Interpreters
- ✓ Natural Languages and Programming Languages

Topic-111. Machine independence

Goal of machine Independence

- ✓ Although the goal of machine independence was largely achieved,
- ✓ Program written for one machine was theoretically used on any machine simply by applying the appropriate compiler
- ✓ The different ways in which machines handle I/O operations.
- ✓ Needs to make minor modifications when program is run on different machines.

Standardization

- ✓ American National Standards Institute and the International Organization for Standardization have adopted and published standards for many of the popular languages.

Language Extensions

- ✓ Compiler designers provide features, called language extensions - that are not part of standard language.
- ✓ If a programmer takes advantage of these features, makes the code machine dependent.

Summary

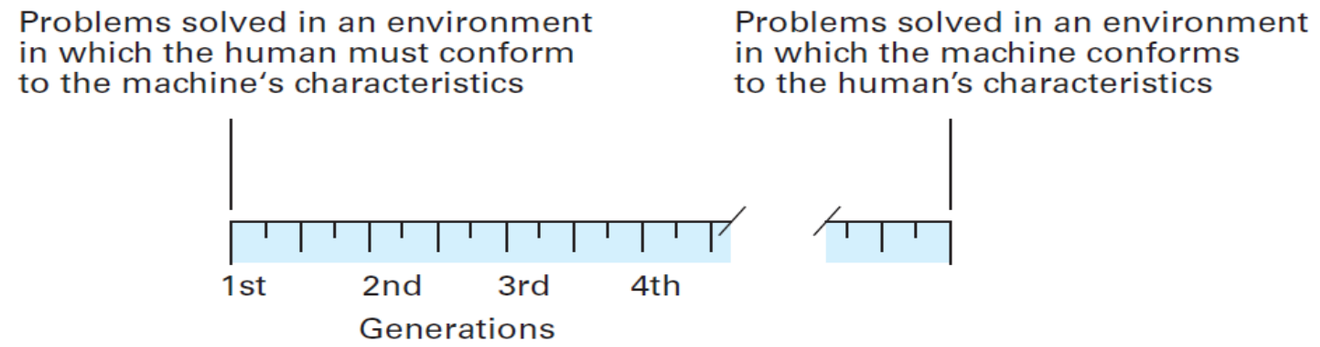
Programming Languages: Machine Independence

- ✓ Discussion on what has been achieved using third generation languages and what remained a dream.
- ✓ Standardization

- ✓ Language extensions

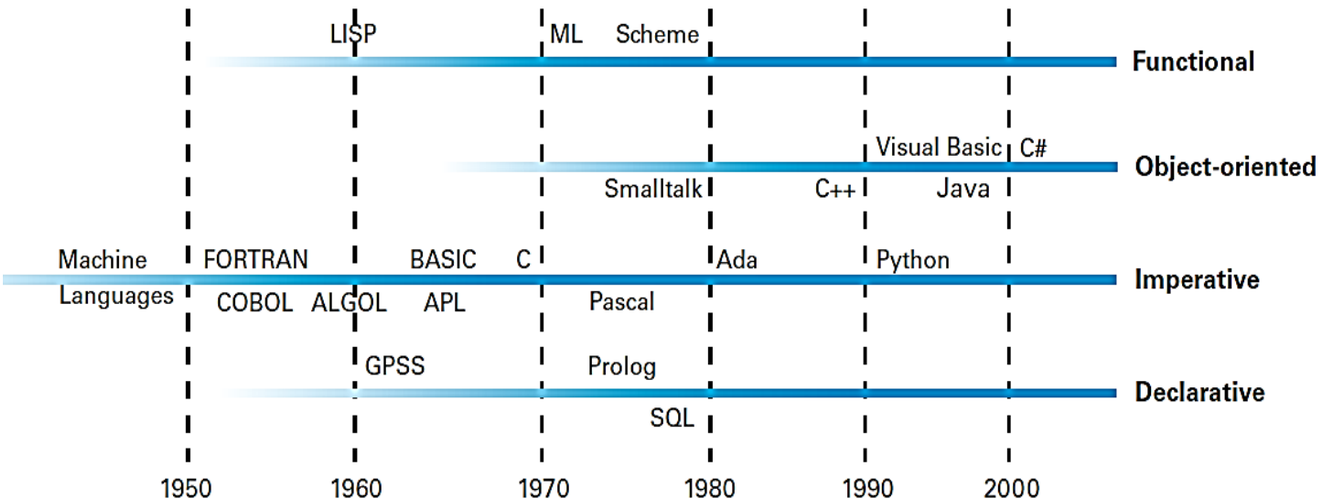
Topic-112. Imperative paradigm

Generations of Programming Languages



Programming Paradigms

- ✓ Alternative Approaches to the programming process



Imperative Paradigm

- ✓ Also known as procedural paradigm
- ✓ Traditional approach of programming process
- ✓ Psuedocode mentioned in previous modules
- ✓ It defines the programming process to be the development of a sequence of commands that, when followed, manipulate data to produce the desired result.
- ✓ Start with finding an algorithm and then express that algorithm as sequence of commands.

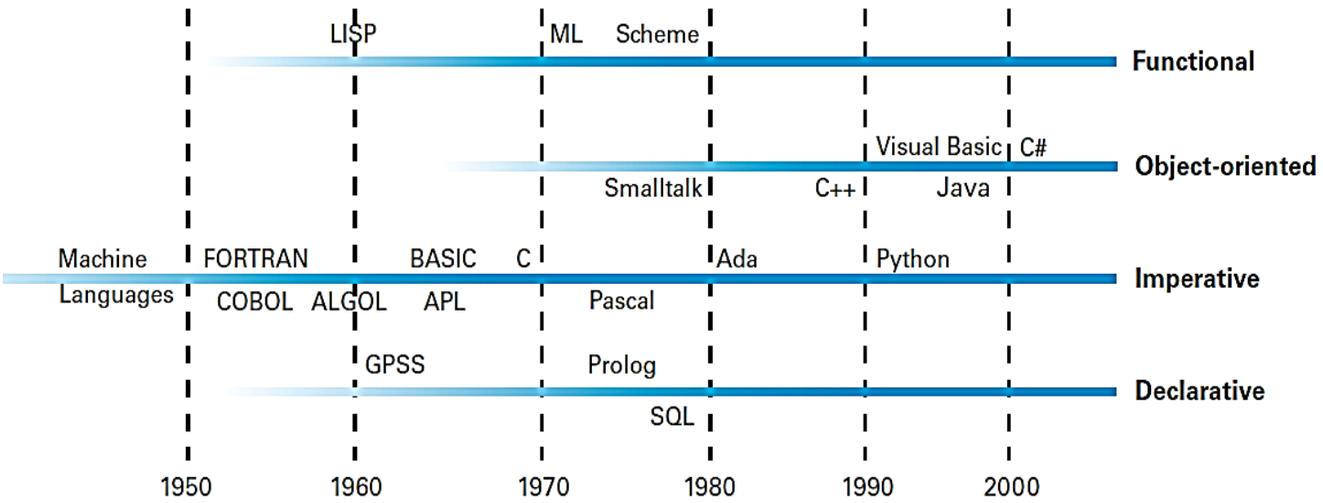
Summary

Programming Languages: Imperative Paradigm

- ✓ Generations of Programming Language
- ✓ Programming Paradigms
- ✓ Imperative Paradigms

Topic-113. Declarative paradigm

Programming Paradigms



Declarative Paradigm

- ✓ Asks a programmer to describe the problem to be solved rather than an algorithm to be followed.
- ✓ Applies a pre-established general-purpose problem-solving algorithm to solve problems presented to it.

Task of Programmer

- ✓ Developing a precise statement of the problem rather than of describing an algorithm for solving the problem.

Obstacle

- ✓ Knowing the generic algorithm and then implementing it.
- ✓ Can be useful for some special purpose software like: Political, economic, environmental etc.

Weather Forecasting

- ✓ Programmer does not need to invent the algorithm.
- ✓ Just need to enter the current weather status

Logic Programming

- ✓ Boost to declarative paradigm with a discovery that the subject of formal logic within mathematics provides a simple problem solving algorithm suitable for use in a general-purpose declarative programming system- emergence to logic programming

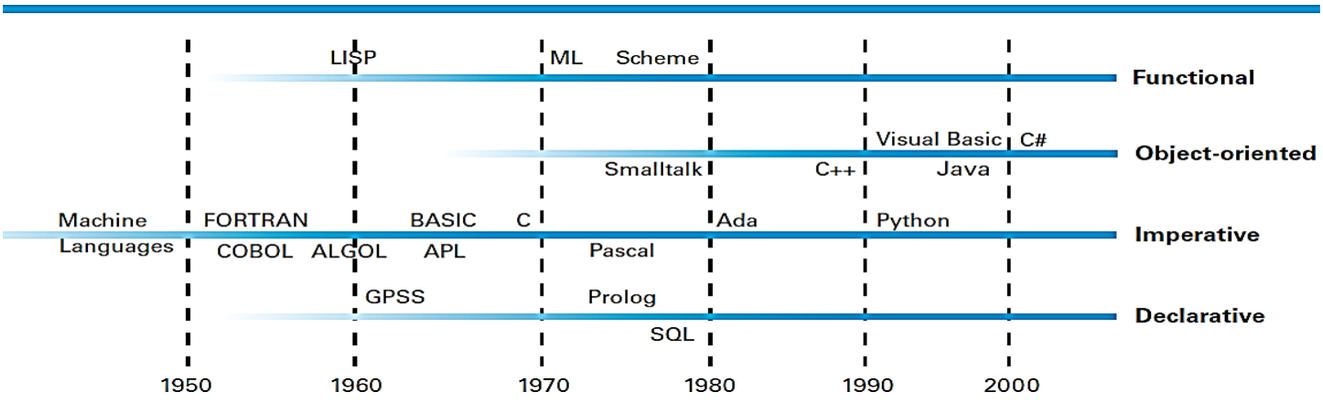
Summary

Programming Languages: Declarative Paradigm

- ✓ Declarative Paradigm
- ✓ Task of programmer
- ✓ Obstacles
- ✓ Logic Programming

Topic-114. Functional paradigm

Programming Paradigms



Functional Paradigm

- ✓ Program is viewed as an entity that accepts inputs and produces outputs.
- ✓ Mathematicians refer to such entities as functions, which is the reason this approach is called the functional paradigm
- ✓ Program is constructed by connecting smaller predefined program units (predefined functions) so that each unit's outputs are used as another unit's inputs in such a way that the desired overall input-to-output relationship is obtained.

Functional Paradigm – Example

Imperative Paradigm

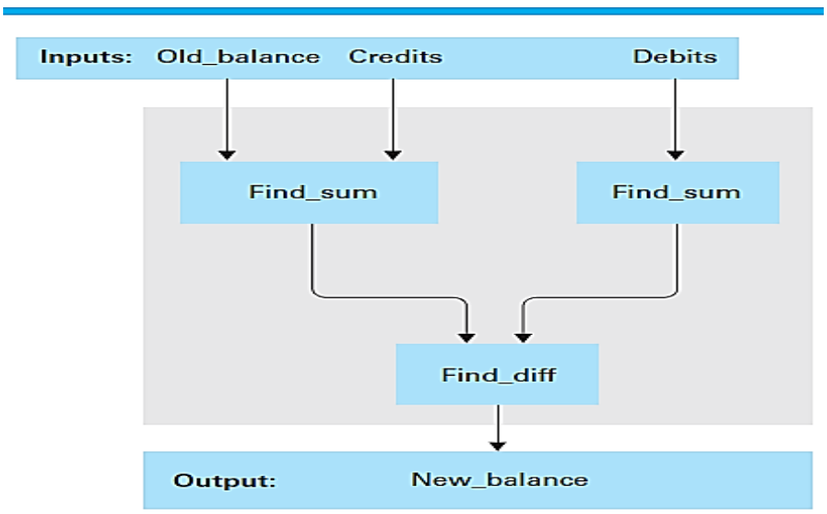
Total_credits = sum of all Credits

Temp_balance = Old_balance + Total_credits

Total_debits = sum of all Debits

Balance = Temp_balance - Total_debits

Functional Paradigm



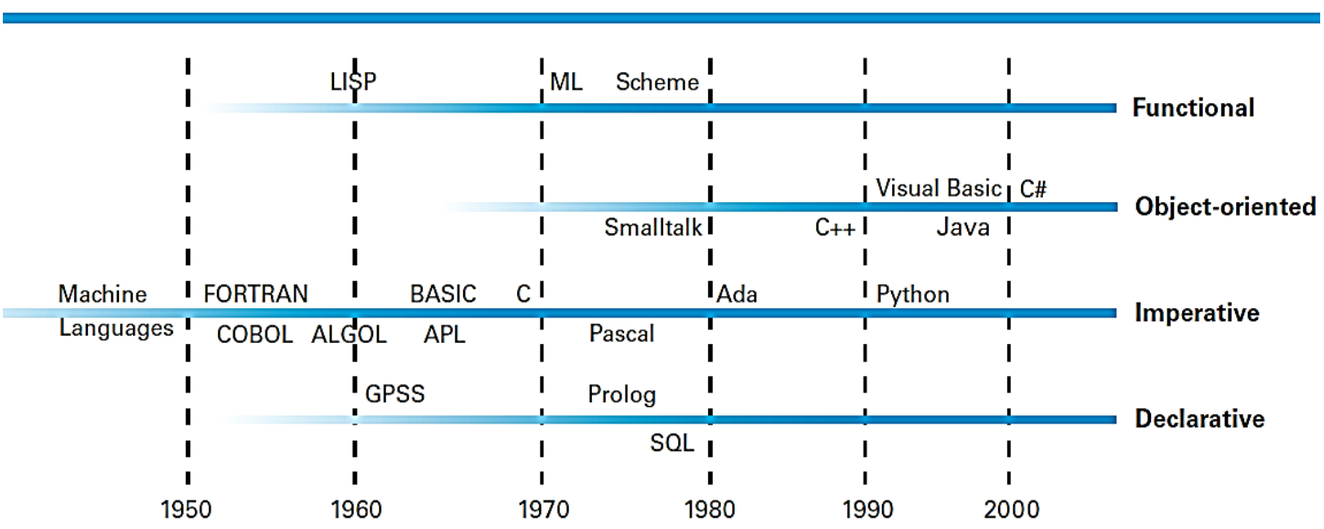
Summary

Programming Languages: Functional Paradigm

- ✓ Functional Paradigm
- ✓ Its Working
- ✓ Example

Topic-115. Object oriented paradigm

Programming Paradigms



Object-Oriented Paradigm

- ✓ Associated with the programming process called object-oriented programming (OOP).
- ✓ software system is viewed as a collection of units, called **objects**

Objects

- ✓ Each object is capable of performing the actions that are immediately related to itself as well as requesting actions of other objects.
- ✓ Together, these objects interact to solve the problem at hand

Developing GUI

- ✓ Icons will be implemented as objects
- ✓ Each object will have its own collection of functions called **Methods**

Difference of List in imperative and OO-Paradigm

- ✓ You need your algorithm for searching sorting etc in imperative, however, in OO-paradigm, all functionality will be available along with the list in terms of object
- ✓ Such Description is called Class

Summary

Programming Languages: Object-oriented Paradigm

- ✓ Object-oriented paradigm
- ✓ Objects
- ✓ Examples
- ✓ Class

Topic-116. Variable and data types

Variable

High-level programming languages allow locations in main memory to be referenced by descriptive names rather than by numeric

addresses. Such a name is known as a **variable**,

Primitive Data Types

- ✓ The data types that are included as primitives in a programming language:
  - ✓ Integer
  - ✓ Float
  - ✓ Character
  - ✓ Boolean

Examples

- ✓ https://www.onlinegdb.com/online_c++_compiler

Summary

Programming Languages: Variable and Data Types

- ✓ Primitive data types
- ✓ Examples

Topic-117. Data structure

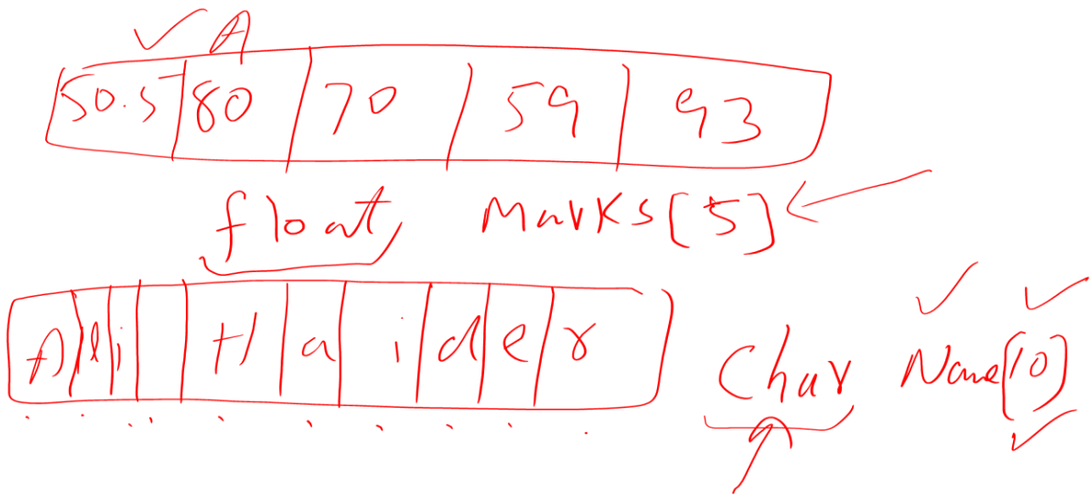
Data Structure

The conceptual shape or arrangement of data.

e.g.

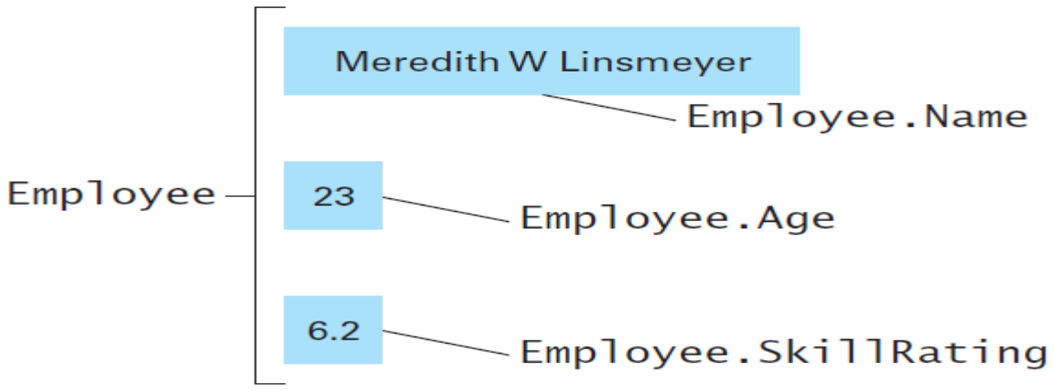
- ✓ Name is a sequence of characters,
- ✓ Student marks in 5 subjects

Example – Array



Record, Structure (heterogeneous array)

```
struct { char Name[25];
int Age;
float SkillRating;
} Employee;
```



More Data Structures

- ✓ Lists
- ✓ Tree
- ✓ Graph
- ✓ and many sub types of these

Summary

Programming Languages: Data Structure

- ✓ Arrays
- ✓ Records
- ✓ Other data structures

Topic-118. Assignment statement

The most basic imperative statement is the **assignment statement**, which requests that a value be assigned to a variable (or more precisely, stored in the memory area identified

Example

```
int marks;
marks = 10;
```

Symbols

Z = X + Y

Z := X + Y;

Summary

Programming Languages: Assignment Statement

- ✓ Practiced with examples

Topic-119. Control structure (if-statement)

Change the execution sequence of the code.

Example – Pass-Fail

```
cout<<"Hello to ITP";
cout<<"Hello to VU";
if(marks>=50)
cout<<"you are passed"
cout<<"bye bye";
```

Summary

Programming Languages: Control Structure (if-statement)

- ✓ Changes the execution flow
- ✓ Practiced with example

Topic-120. Control Structures (if-statement examples)

Example – Scholarship award

```
float CGPA = 3.5;
if (CGPA>=3.0)
cout<<"Give scholarship";
else
cout<<"Sorry you donot
deserve for scholarship";
```

Summary

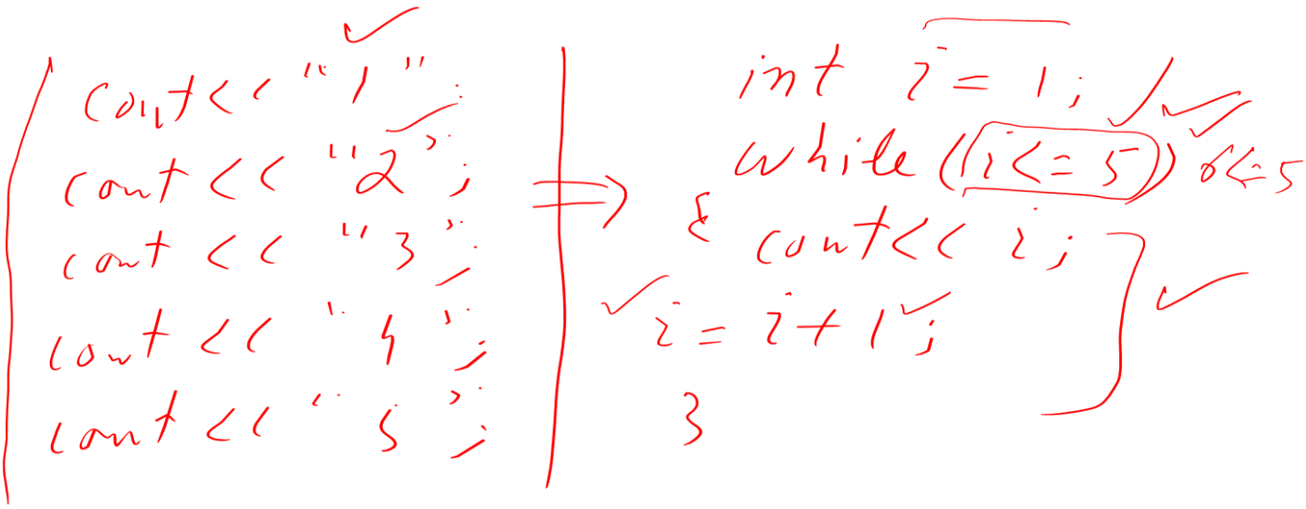
Programming Languages: Control Structure (if-statement examples)

- ✓ Changes the execution flow
- ✓ Practiced with example

Topic-121. Control structures (loops)

- ✓ Repeatedly executes a set of instructions

Printing counting



Summary

Programming Languages: Control Structure (Loops)

- ✓ Repeats set of instructions
- ✓ Practiced with example

Topic-122. Programming concurrent activities

Concurrent Processing

Simultaneous execution of multiple activations is called **parallel processing** or **concurrent processing**.

Scenario

Suppose you have been asked to produce animation for an action computer game

True Parallel Processing

Possible when multiple CPU core process each activation

When one CPU, illusion can be created using multiprogramming systems.

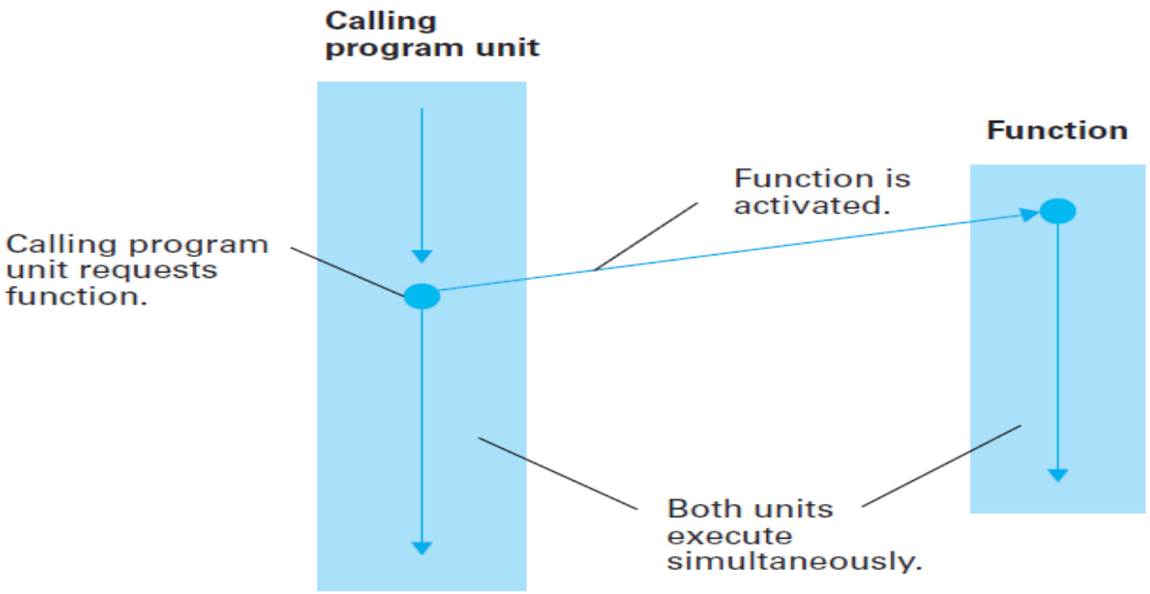
Activation

Different methodologies and naming conventions in different programming languages.

- ✓ Task in Ada
- ✓ Thread in Java

Procedure

Creating main program which creates activations/threads/tasks



Summary

Programming Languages: Programing Concurrent Activities

- ✓ Concurrent Processing/Parallel processing
- ✓ Scenario
- ✓ How it is achieved.

Topic-123. Arithmetic operators examples

Operators understanding and practicing online

- +
- -
- *
- /
- %

Summary

Programming Languages: Arithmetic Operator examples

- ✓ Operators
- ✓ Examples.

Topic-124. Relational operators examples

Operators understanding and practicing online

- <
- <=
- >
- >=
- ==
- !=

Summary

Programming Languages: Relational Operator examples

- ✓ Operators
- ✓ Examples.

Topic-125. Logical operators examples

Operators understanding and practicing online

&&

||

!

Summary

Programming Languages: Logical Operator examples

- ✓ Operators
- ✓ Examples.

Topic-126. Software engineering discipline

Scenario

- ✓ Constructing a Commercial building:
- ✓ Floors = 5
- ✓ Size = 50 feet by 90 feet.

Estimations???

- ✓ How can you estimate the cost in time, money, and other resources to complete the project?
- ✓ How can you divide the project into manageable pieces?
- ✓ How can you ensure that the pieces produced are compatible?
- ✓ How can those working on the various pieces communicate?
- ✓ How can you measure progress?
- ✓ How can you cope with the wide range of detail

Such questions need to be answered for SE projects as well!

Wrong Estimations!

- ✓ Could lead to cost overruns
- ✓ Late delivery of products.
- ✓ Dissatisfied customers

Differences b/w Software Engineering and Engineering

- ✓ Using off-the-shelf components (making new vehicle does not require to develop engine)
- ✓ Quantitative techniques called metrics e.g. measuring complexity of software

Computer-aided SE

- ✓ Continuing to streamline the software development process.
- ✓ CASE tools:
  - ✓ Cost estimation
  - ✓ Project scheduling
  - ✓ Personnel allocation
  - ✓ Progress Monitoring

Computer-aided SE

- ✓ CASE tools:
  - ✓ Writing and organizing documentation
  - ✓ Prototyping
  - ✓ Interface design
  - ✓ Writing and debugging codes.

Integrated Development Environments (IDE)

- ✓ Combine tools for development:
  - ✓ Editors
  - ✓ Compilers

- ✓ Debugging tools

Summary

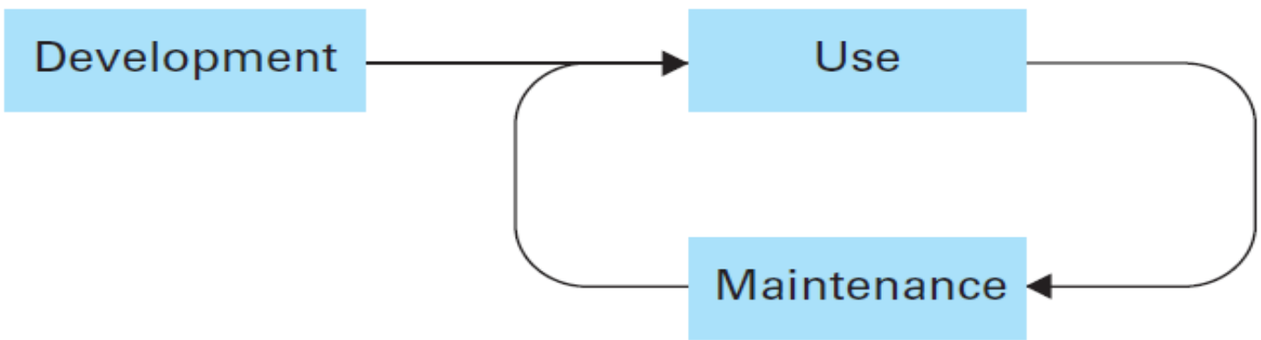
Software Engineering

Discipline

- ✓ Scenario.
- ✓ Estimations
- ✓ Drawback of wrong estimations
- ✓ SE and other Engineering disciplines
- ✓ CASE
- ✓ IDEs

Topic-127. Software life cycle

- ✓ It's the most fundamental concept in Software engineering.



- ✓ Maintenance vs repair

Why Software is moved to maintenance

- ✓ Errors are discovered
- ✓ Changes in software application occurs
- ✓ Changes done in previous modification introduce the errors.

Process of maintenance

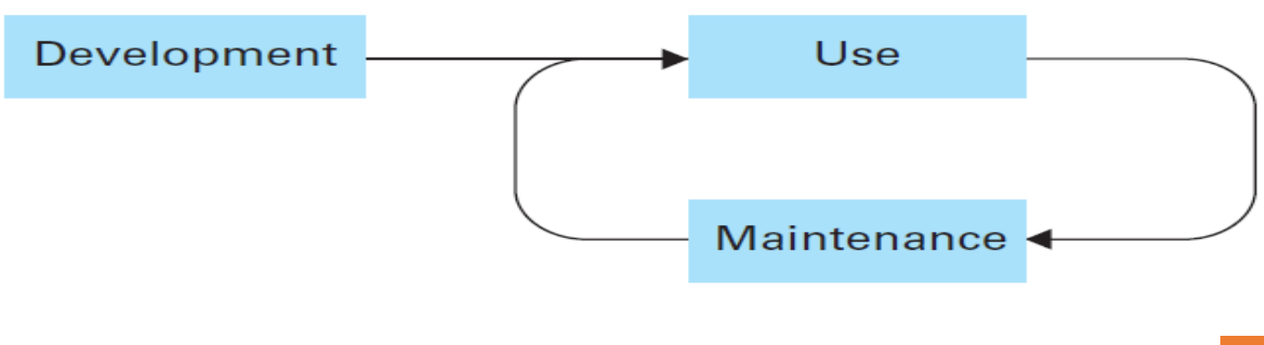
- ✓ Studying the underlying program and documentation.

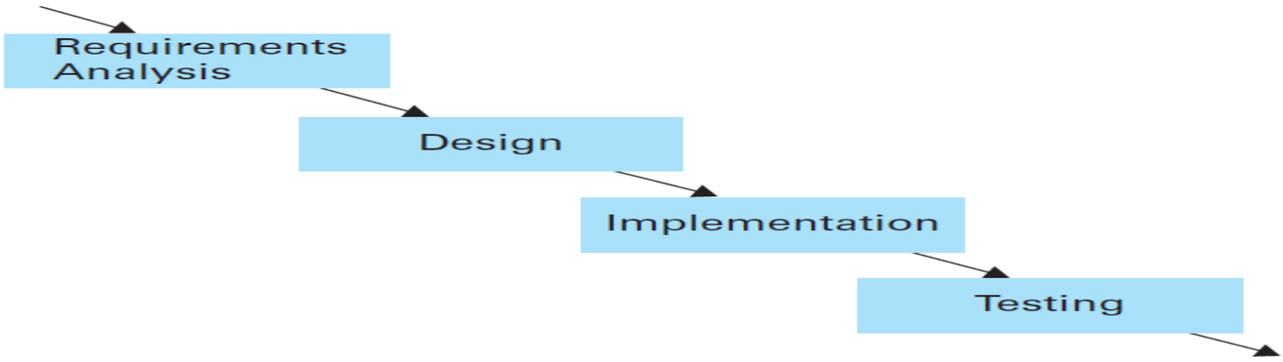
Summary

Software Life Cycle

- ✓ Software Life Cycle
- ✓ Maintenance vs repair
- ✓ Why software is added to maintenance phase
- ✓ Process of maintenance

Topic-128. Requirement analysis phase





Goal

What services the proposed system will provide, to identify any conditions (time constraints, security, and so on) on those services,

and to define how the outside world will interact with the system.

Inputs from Stakeholders

- ✓ Future users.
- ✓ Other having legal and financial interests
- ✓ When user is entity, company or government agency, they hire software developer

Commercial off-the-shelf (COTS)

- ✓ Software developed for the mass market
- ✓ perhaps to be sold in retail stores
- ✓ downloaded via the Internet,
- ✓ In this case, market study is required.

Requirement Analysis Process

- ✓ Compiling the needs of user.
- ✓ Negotiating with project stakeholders on trade-offs between wants, needs, cost, and feasibility

Software Requirement Specification (SRS)

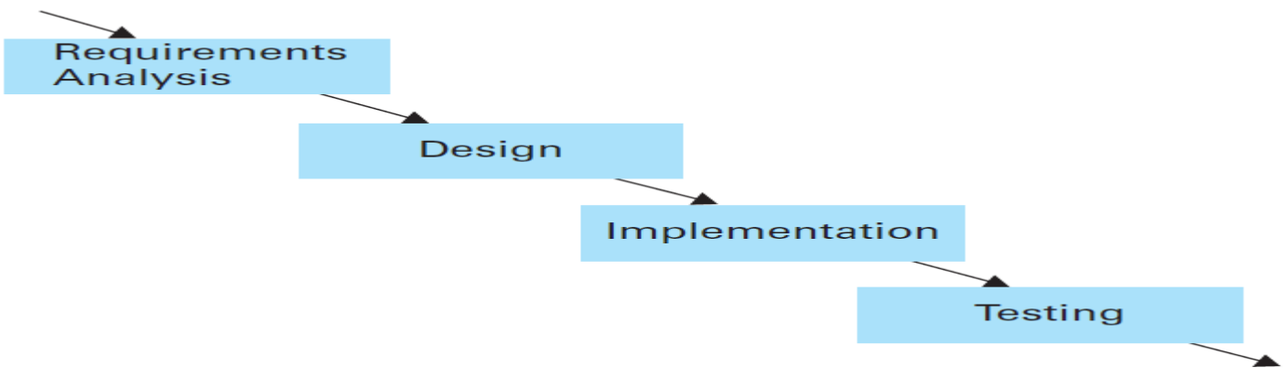
- ✓ Requirements are recorded in this document
- ✓ This document is a written agreement between all parties concerned

Summary

Requirement Analysis Phase

- ✓ SDLC
- ✓ Requirement Analysis
- ✓ Goal and process
- ✓ Stakeholders
- ✓ COTS
- ✓ SRS

Topic-129. Design phase



RA vs Design

Requirements analysis provides a description of the proposed software

product, Design involves creating a plan for the construction of the proposed system.

RA identifies problems, whereas, design finds solution

Requirements analysis decides what a system will do, and design identifies how the system will do it

Design Outcome

- ✓ Internal structure of the software system is established.
- ✓ Detailed description of the software system's structure that can be converted into programs

Office building example

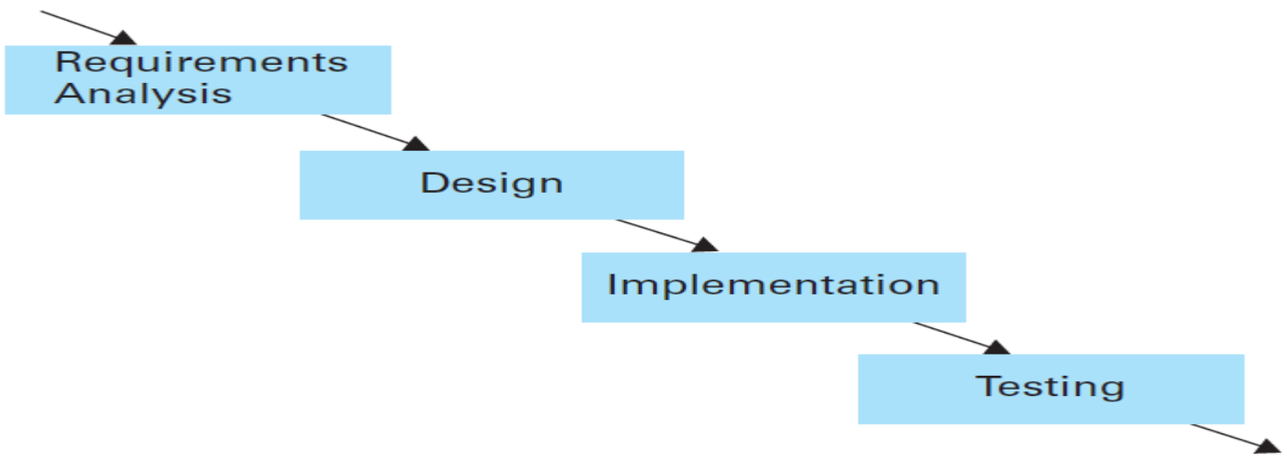
- ✓ Structural plans
- ✓ Making designs that can be converted into programs.
- ✓ Blueprints describing the proposed building at various levels of detail
- ✓ In Software design: notational system and many modeling and diagramming methodologies

Summary

Design Phase

- ✓ Requirement analysis vs design
- ✓ Design outcome
- ✓ Analogy with building design

Topic-130. Implementation phase



Goal

Actual writing of programs, creation of data files, and development of databases.

Building construction analogy

Software Analyst vs programmer

Requirement analysis, design, Implementation

However, no

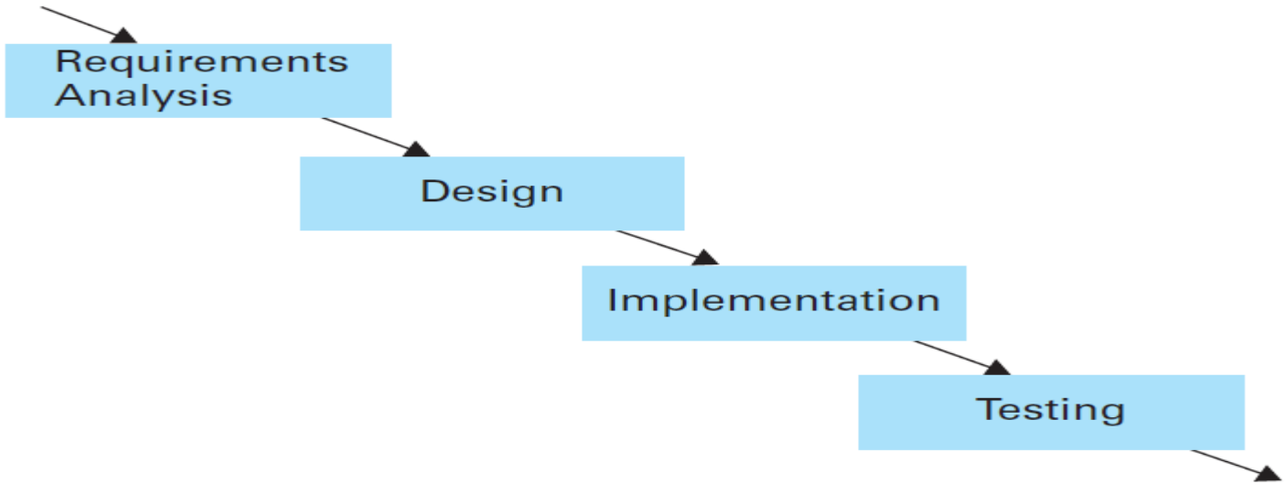
central authority controlling the use of terminology throughout the computing community

Summary

Implementation Phase

- ✓ Goal
- ✓ Software Analyst and programmer

Topic-131. Testing phase



Testing in traditional software development

Testing remained to be considered just to debug programs and confirming that the software is compatible to SRS.

Modern Testing

Programs are not the only artifacts that are tested during the software development process. Indeed, the result of each intermediate step in the entire development process should be “tested” for accuracy.

Testing should not be considered as separate step:

- ✓ Requirements analysis and confirmation
- ✓ Design and validation
- ✓ Implementation and testing.”

Need for better testing methodologies!

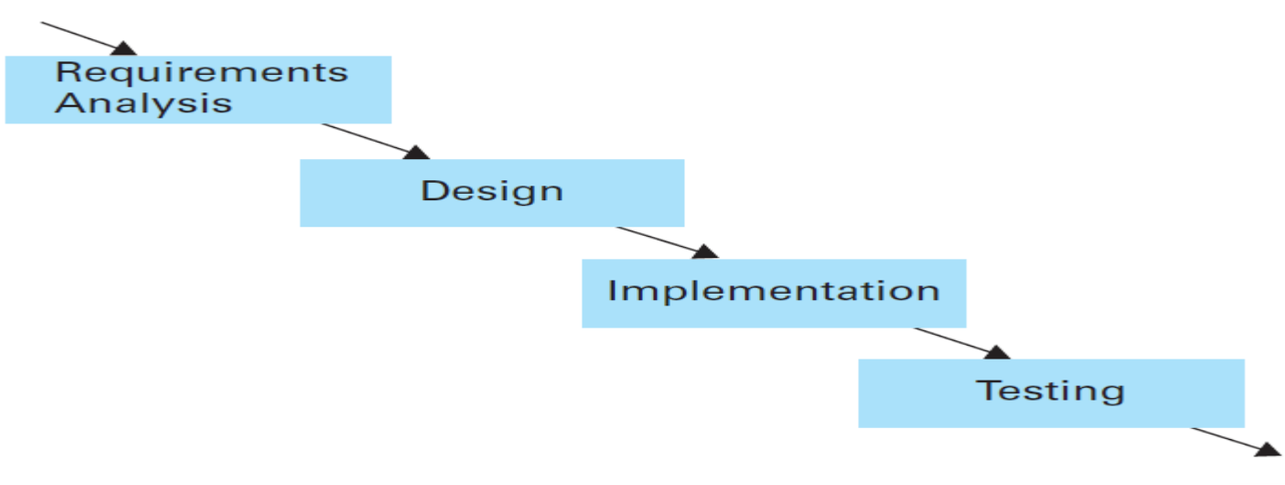
- ✓ Large software systems continue to contain errors, even after significant testing.
- ✓ Many of these errors may go undetected for the life of the system
- ✓ Others may cause major malfunctions

Summary

Testing Phase

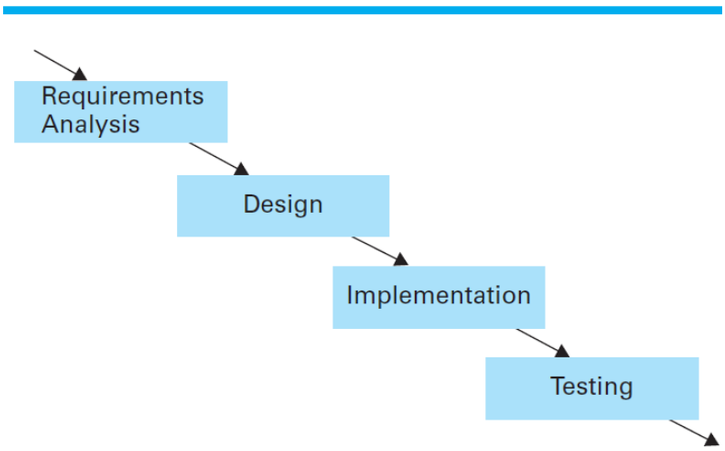
- ✓ Traditional testing approach
- ✓ Modern software testing
- ✓ Current issues

Topic-132. Software engineering methodologies-1



Water Fall Model

Entire preceding phase need to be completely done before the start of next phase.



Incremental Model

- ✓ Desired software system is constructed in increments the first being a simplified version of the final product with limited functionality.
- ✓ Example: Student add, view, course addition

Iterative Model

- ✓ Sometimes confused with Incremental model.
- ✓ **Incremental model:** extending each preliminary version of a product into a larger version iterative model.
- ✓ **Iterative model:** refining each version creating a working prototype first.

Rational Unified Process

- ✓ RUP is an example of iterative techniques, created by Rational Software Corporation, now owned by IBM.
- ✓ Redefines the steps in the development phase of the software life cycle
- ✓ RUP is widely applied now a days in software industry.
- ✓ Non-proprietary version is Unified Process.

Summary

Software Engineering Methodologies – I

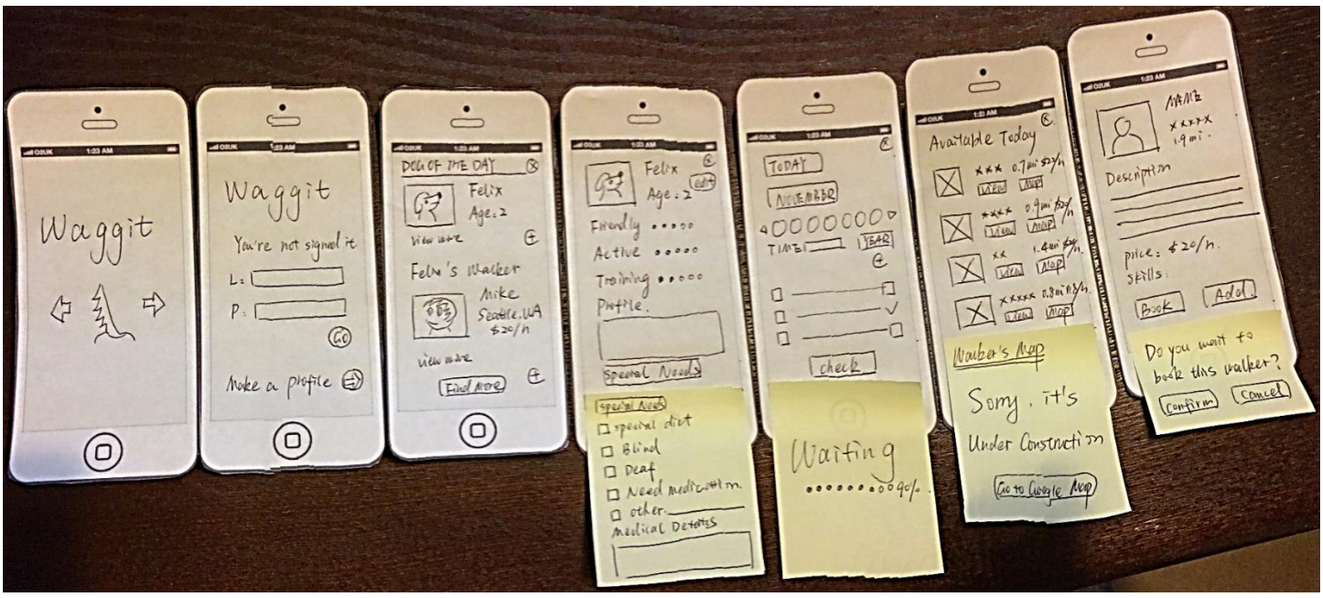
- ✓ Water Fall model
- ✓ Incremental model
- ✓ Iterative model
- ✓ Rational Unified Process
- ✓ Unified Process

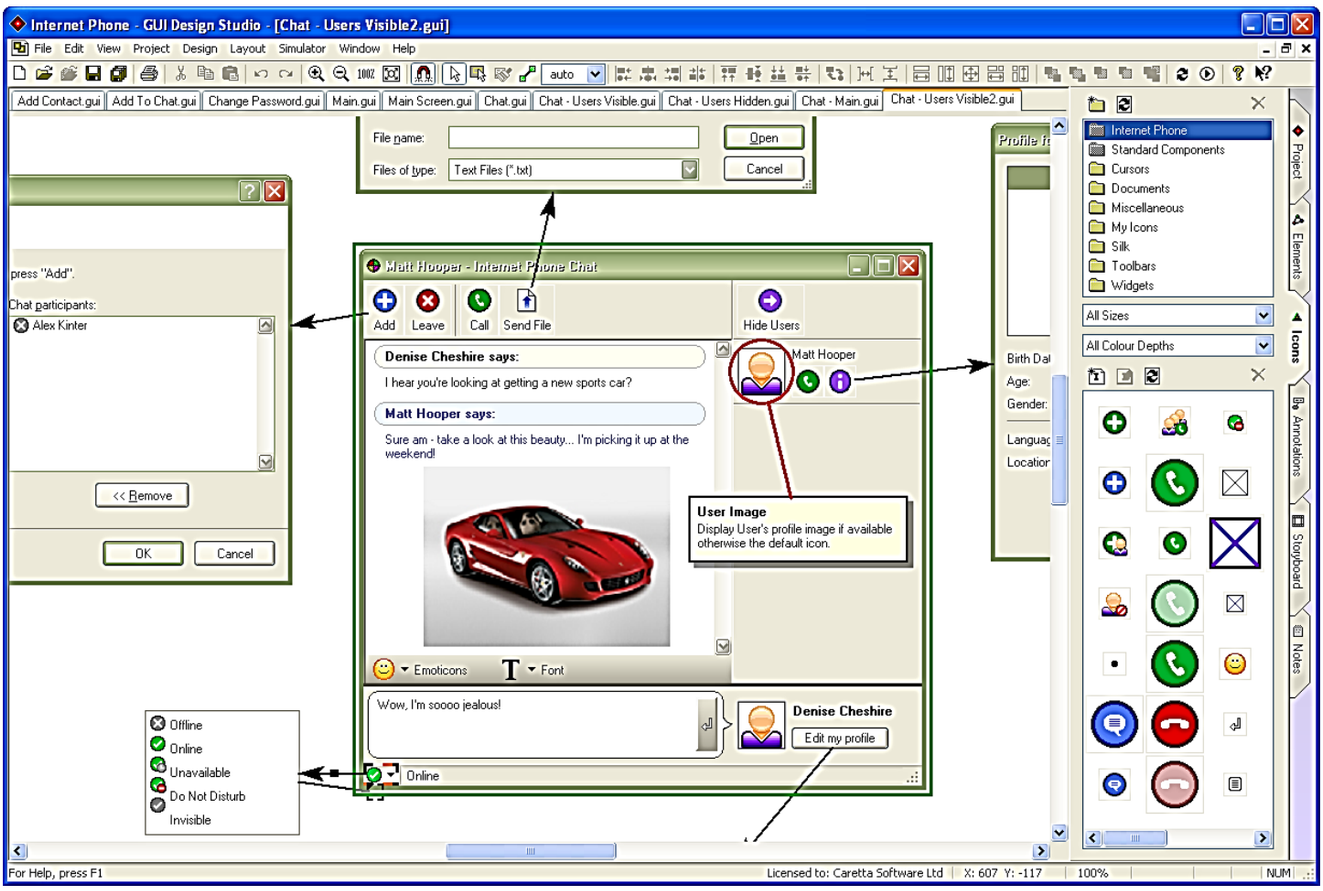
Topic-133. Software engineering methodologies-2

Prototype

- ✓ Incomplete version of the proposed system built and evaluated in Iterative and Incremental models.

Paper - Prototype





Evolutionary Prototyping

- ✓ In the case of the incremental model, initial prototypes evolve into the complete, final system. This process is called evolutionary prototyping.

Throwaway Prototyping

- ✓ In a more iterative situation, the prototypes may be discarded in favor of a fresh implementation of the final design. This approach is known as throwaway prototyping.

Rapid Prototyping

- ✓ Simple example of the proposed system is quickly constructed in the early stages of development.
- ✓ Demonstration version

Open Source Development

- ✓ Purpose is to produce the free software.
- ✓ A single author writes the initial version.
- ✓ Source code and documentation is shared via internet where others can contribute.

Agile methods

- ✓ pronounced shift from the waterfall model is represented by the collection of methodologies known as agile methods.
- ✓ proposes early and quick implementation on an incremental basis

Extreme Programming

- ✓ One example of an agile method is extreme programming (XP).
- ✓ Software is developed by a team of less than a dozen individuals basis, by means of repeated daily cycles and helping each other.
- ✓ Can be evaluated by project stakeholders, at different stages.

Summary

Software Engineering Methodologies – II

- ✓ Prototype
- ✓ Open Source Development
- ✓ Agile Methods
- ✓ Extreme Programming

Topic-134. Modular implementation

Modularity

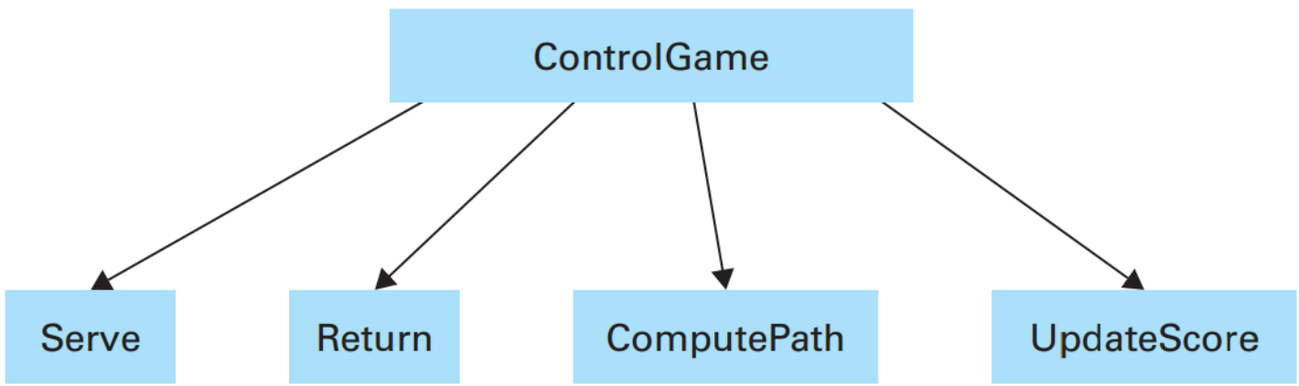
- ✓ To modify software, one should understand the software, difficult in small programs and nearly impossible for large software.
- ✓ Module wise implementation – division of software into manageable units, generically called **modules**

What are modules?

- ✓ Imperative programming paradigm – functions
- ✓ Object oriented paradigm – Objects

Example – imperative Paradigm

- ✓ Serve () – Speed, Direction, players characteristics
- ✓ ComputePath() – hit net, where it bounce,
- ✓ Return() – will it be returned, next speed, direction
- ✓ UpdateScore()



Summary

Modular Implementation

- ✓ Modularity
- ✓ Modules
- ✓ Example – Imperative paradigm

Topic-135. Coupling

Modularity advantage

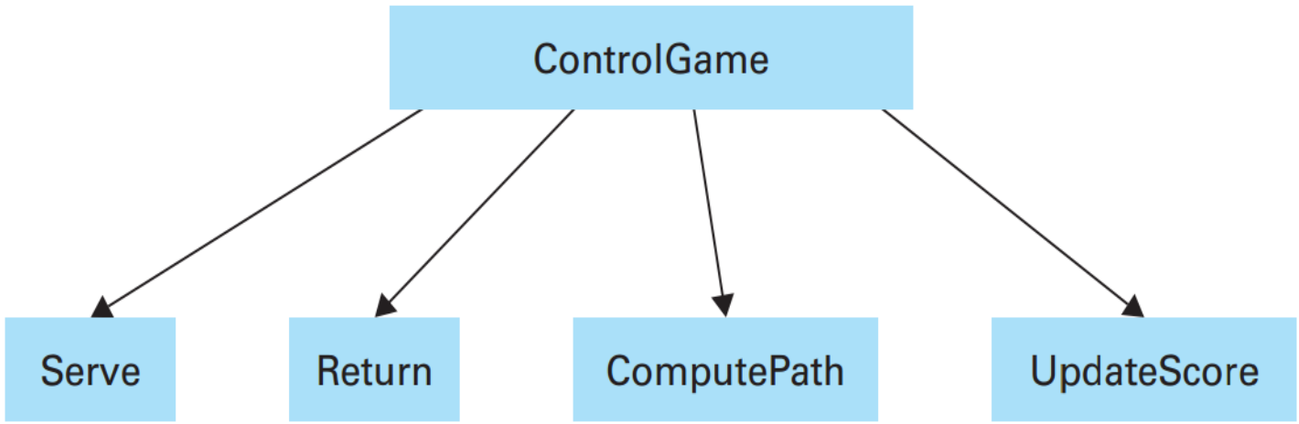
- ✓ Any modification will be applied to few of the modules
- ✓ Assumption: Changes in one module will not affect other modules.
- ✓ Goal: maximize the independence, minimize linkage between modules known as: **intermodule coupling**

Intermodule Coupling

- ✓ One metric to measure the complexity - expense of maintaining the software is to measure the intermodule coupling

Control Coupling

- ✓ Occurs when a module passes control of execution to another, as in a function call.



Data Coupling

- ✓ Refers to the sharing of data between modules. If two modules interact with the same item of data, then modifications made to one module may affect the other

Explicit passing

- ✓ Passing by parameters. ControlGame will tell the function Serve() which player's characteristics are to be simulated when it calls Serve and that the function Serve() will report the ball trajectory to ControlGame when Serve() has completed its task.

Global data

- ✓ Data items that are automatically available to all modules throughout the system.

Summary

Coupling

- ✓ Intermodule Coupling
- ✓ Control Coupling
- ✓ Data Coupling

Topic-136. Cohesion

- ✓ The term cohesion refers to this internal binding or, in other words, the degree of relatedness of a module's internal parts.

Goal

- ✓ Low intermodule coupling.
- ✓ High intramodule cohesion

Logical Cohesion

- ✓ This is the cohesion
  within a module induced by the fact that its internal elements perform activities logically similar in nature. Communication module – obtaining data and reporting results

Functional Cohesion

- ✓ All the parts of the module are focused on the performance of a single activity results.
- ✓ Can be increased by isolating subtasks in other modules and then using these modules as abstract tools

Summary

Cohesion

- ✓ Goal
- ✓ Logical Cohesion
- ✓ Functional Cohesion

Topic-137. Information hiding

- ✓ Refers to the restriction of information to a specific portion of a software system.
- ✓ Information should be interpreted in a broad sense such as: data, the type of data structures used, encoding systems, the internal compositional structure of a module etc

Why we need it

- ✓ To reduce unnecessary dependencies or effects on other modules.
- ✓ for example, a module does not restrict the use of its internal data from other modules, then that data may become corrupted by other modules

Realization

- ✓ Design Goal
- ✓ Implementation Goal

Design Goal

- ✓ A module should be designed so that other modules do not need access to its internal information
- ✓ Example: maximizing cohesion and minimizing coupling

Implementation Goal

- ✓ A module should be implemented in a manner that reinforces its boundaries.
- ✓ Examples: use of local variables, applying encapsulation, and using well defined control structures

Summary

Information Hiding

- ✓ Why we need it.
- ✓ Design Goal
- ✓ Implementation Goal

Topic-138. Components

Why we need it?

- ✓ We talked about off-the-shelf building blocks from which large software can be constructed.
- ✓ Modular approach just promises?
- ✓ Object-oriented paradigm is helping as objects are self- contained

Prefabricated templates

- ✓ C++ has standard template library
- ✓ Java has Java Application Programming Interface (API)
- ✓ C# programmers have access to .NET Framework Class Library

Components are not just objects!

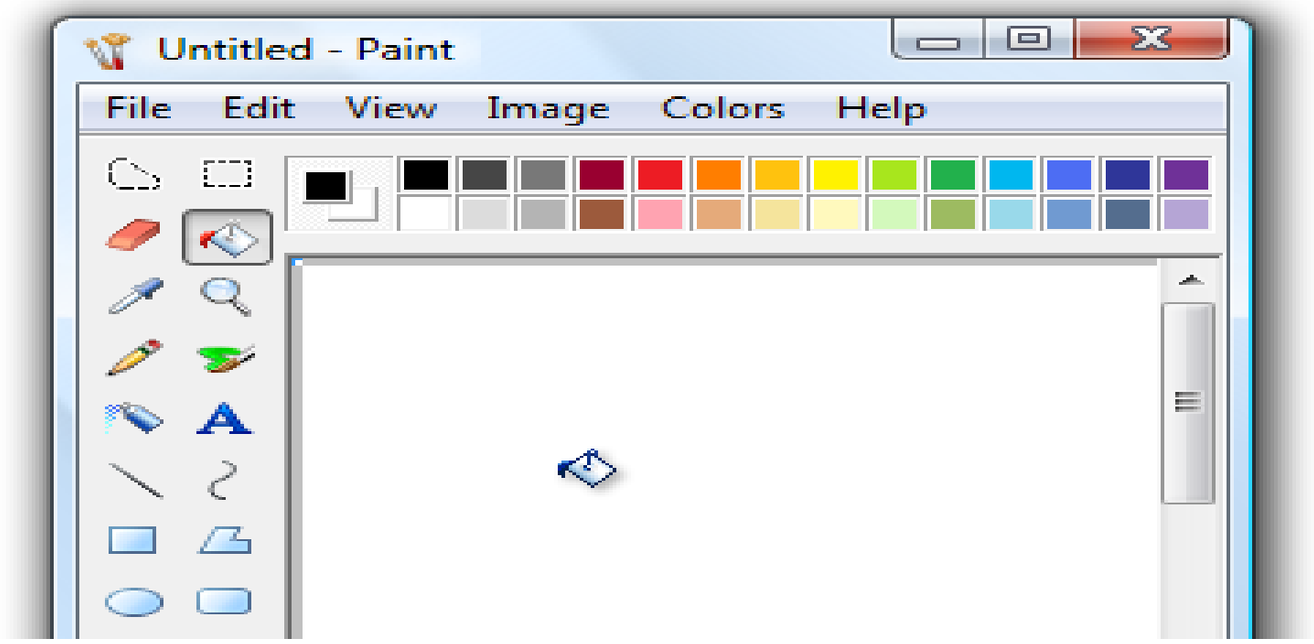
- ✓ Reusable unit of software. In practice, most components are based on the object-oriented paradigm and take the form of a collection of one or more objects that function as a self-contained unit.

Component Architecture

- ✓ also known as component-based software engineering) in which the traditional role of a programmer is replaced by a **component assembler** who constructs software systems from prefabricated components

Example

- ✓ In development interfaces, icons are displayed.
- ✓ the methodology of a component assembler is to select pertinent components from collections of predefined components and then connect them, with minimal customization, to obtain the desired functionality.



More Examples

- ✓ Facebook when executed on a smartphone may use the components of the contact application to add all Facebook friends as contacts.
- ✓ The telephony application, may also access the contact components to lookup the caller of an incoming call

Summary

Components

- ✓ Why we need it.
- ✓ Components Architecture
- ✓ Prefabricated Templates
- ✓ Examples

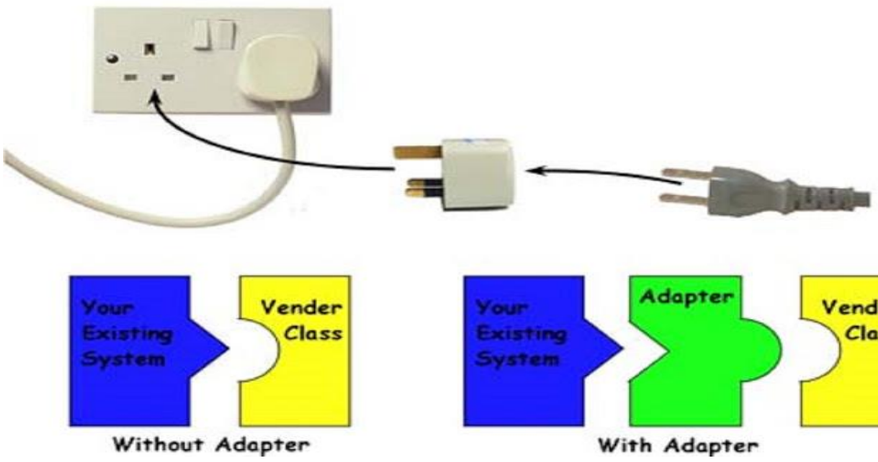
Topic-139. Design patterns

Why we need it?

- ✓ Pre-developed model for solving a recurring problem in software design.
- ✓ You are not the first one working on the problem!

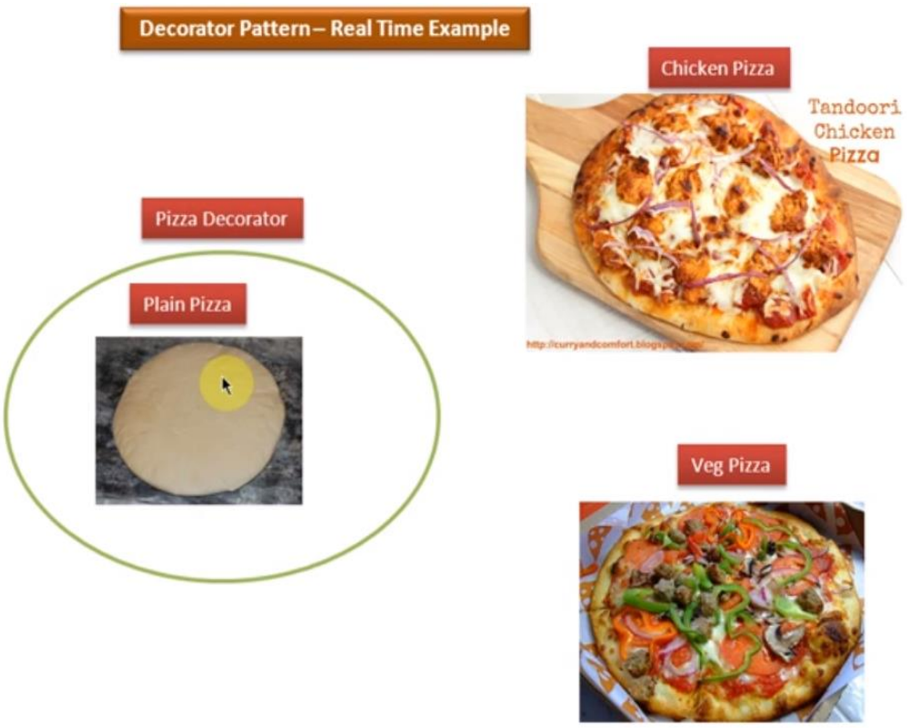
Adapter Pattern

- ✓ A prefabricated module may have the functionality needed to solve the problem at hand but may not have an interface that is compatible with the current application.
- ✓ Adapter pattern provides a standard approach



Decorator Pattern

- ✓ Used when designing a system that performs different combinations of the same activities depending on the situation at the time.
- ✓ Explosion of options.
- ✓ Decorator pattern provides a standardized way of implementing such systems that leads to a manageable solution



Goal

- ✓ Identification of recurring problem, creation and cataloging of design patterns for solving them is an ongoing process in software engineering.
- ✓ Goal is not to identify the solution, the goal is to identity the best solution flexible for future changes!

Summary

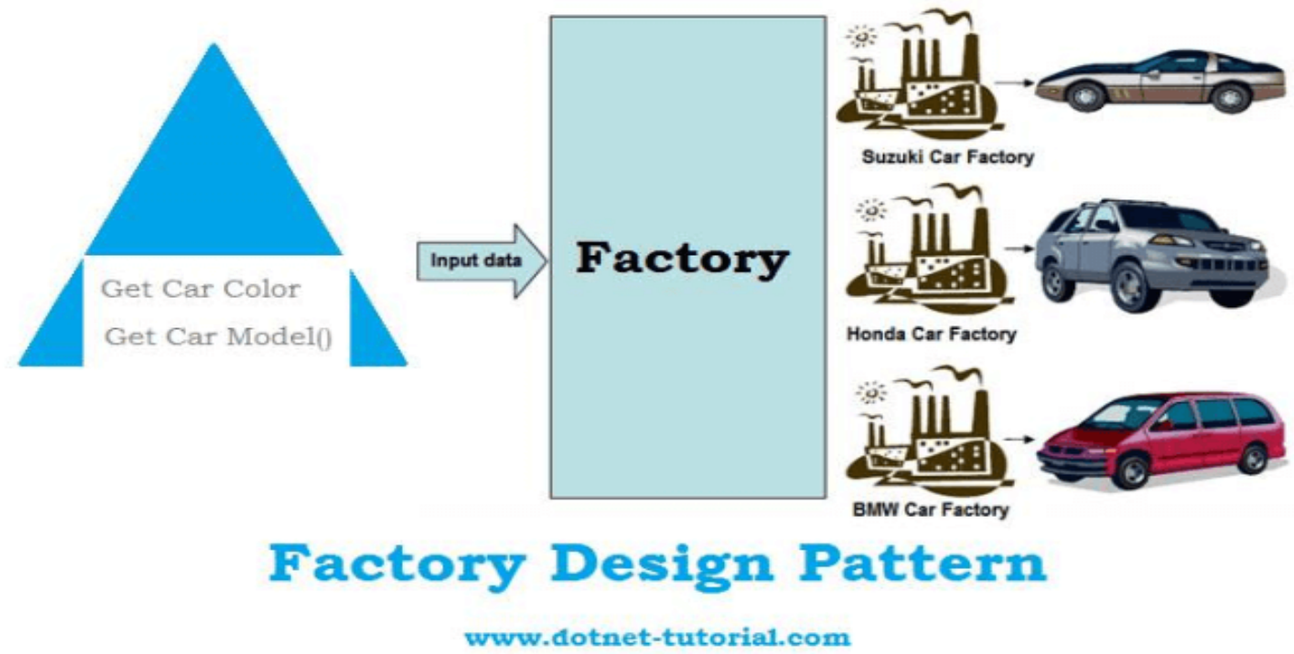
Design Pattern

- ✓ Why we need it.
- ✓ Adapter Pattern
- ✓ Decorator Pattern
- ✓ Goal

Topic-140. Design patterns examples

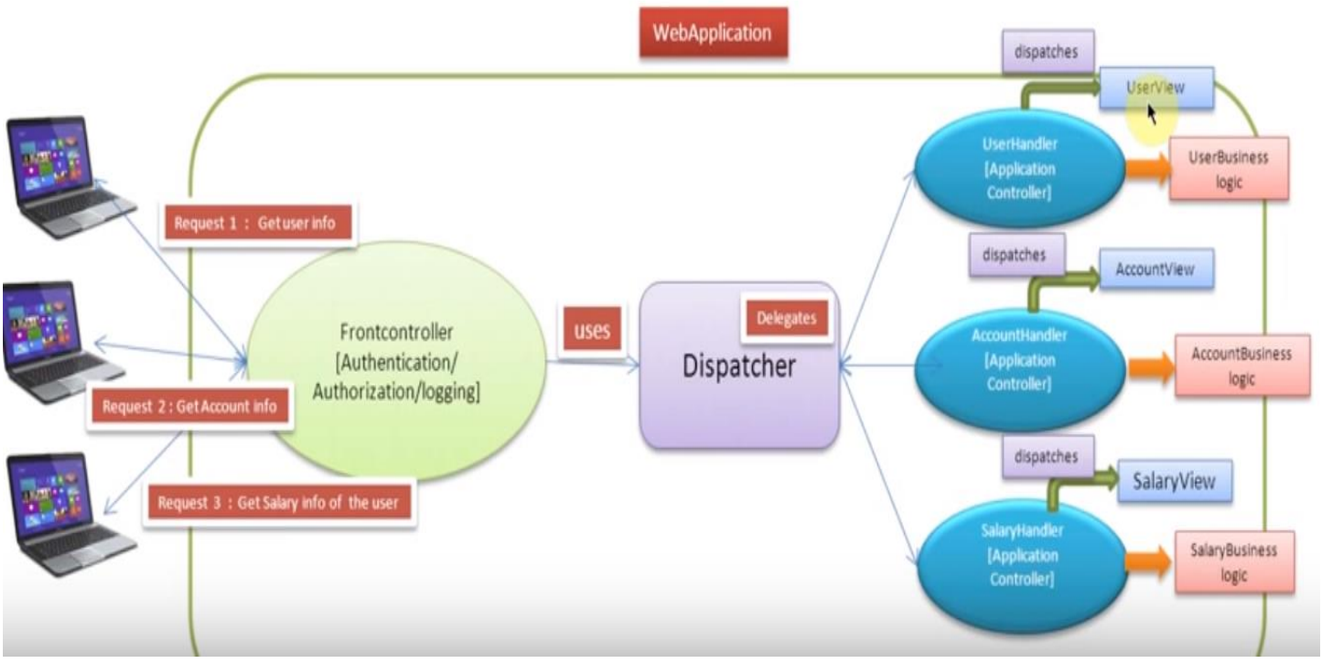
Factory Design Pattern

- ✓ **Creation of loosely coupled system.**



Front Controller Pattern

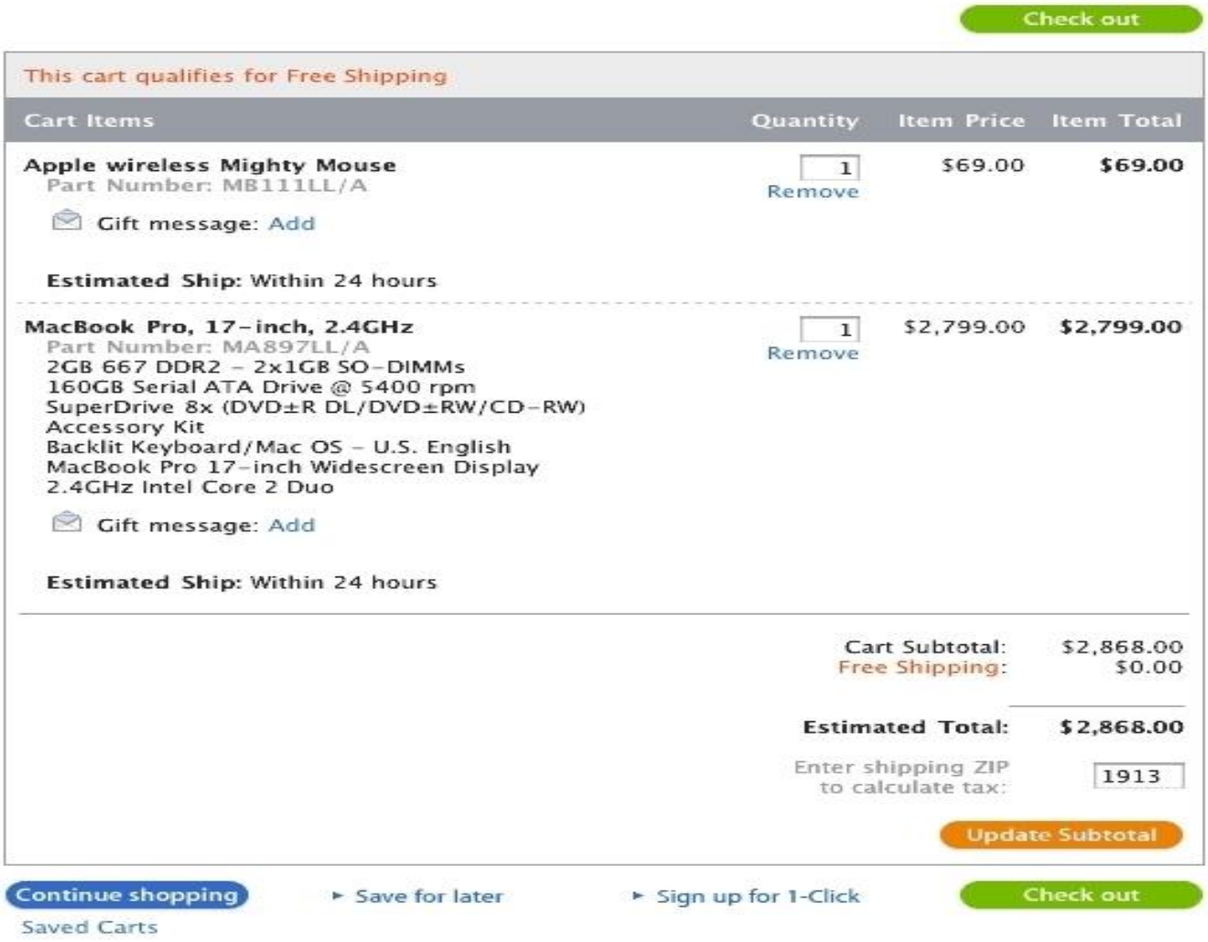
- ✓ Centralized request handling mechanism



Shopping cart design pattern

Usage when:

- ✓ to buy more than one product.
- ✓ To buy more than one instance of a product.
- ✓ Want to return later to carry shopping.
- ✓ Want to return later for payment.
- ✓ Do not use when only one product to sale.
- ✓ Do not use when only one product can be sold.



Summary

Design Patterns Examples

- ✓ Factory Design Pattern
- ✓ Front Controller Pattern
- ✓ Shopping Cart Pattern

Topic-141. Scope of quality assurance

Early years of Computing

- ✓ Software Quality was just meant to identify programming error that occurred during implementation

Contemporary scope

- ✓ Improvement of software engineering procedures
- ✓ Development of training programs that in many cases lead to certification
- ✓ Establishment of standards on which sound software engineering can be based.

Quality Standards

- ✓ ISO, IEEE, and ACM establishing standards for assessing quality control within software development companies
- ✓ ISO 9000: series of standards, which address numerous industrial activities such as design, production, installation, and servicing.

Software Quality Assurance (SQA)

- ✓ Software contractors now require that the organizations meet such standards. They have then SQA groups.
- ✓ Example, in waterfall model, SQA approves SRS before design can start or approves design before implementation starts.

Scope of Quality Assurance

- ✓ Early generations to now.
- ✓ Quality standards

- ✓ SQA

Topic-142. Software testing

SQA and Testing

- ✓ SQA is a now recognized as a subject dealing entire development process, testing is concerned about the developed software/programs

Can we test all possibilities?

- ✓ In simple programs, there may be billions of different paths that could potentially be traversed.
- ✓ Testing all paths in a complex program is nearly impossible.
- ✓ What to do then?

Pareto Principle

- ✓ Economist and sociologist Vilfredo Pareto (1848−1923) invented that small part of Italy's population controlled most of Italy's wealth

Pareto Principle in Software

- ✓ Small number of modules within a large software system tend to be more problematic than the rest
- ✓ Results can often be increased most rapidly by applying efforts in a concentrated area

Basis Path Testing

- ✓ Develop a set of test data that insures that each instruction in the software is executed at least once.
- ✓ Glass-box testing: software tester is aware of the interior structure of the software and uses this knowledge

Black-box Testing

- ✓ Refers to tests that do not rely on knowledge of the software's interior composition.
- ✓ Black-box testing is performed from the user's point of view.
  uses this knowledge

Boundary value analysis

- ✓ Identifying ranges of data, called equivalence classes, over which the software should perform in a similar manner and then testing the software on data close to the edge of those ranges

Beta Testing

- ✓ Preliminary version of the software is given to a segment of the intended audience with the goal of learning how the software performs in real-life situations before the final version of the product is released to the market

Alpha Testing

- ✓ Similar testing performed at the developer's site is called alpha testing.

Summary

Software Testing

- ✓ SQA and Testing
- ✓ Pareto Principle
- ✓ Basis path testing
- ✓ Black-box Testing
- ✓ Boundary Value Testing
- ✓ Beta Testing
- ✓ Alpha Testing

Topic-143. Software documentation

Why we need this?

- ✓ A software system is of little use unless people can learn to use and maintain it.
- ✓ user documentation
- ✓ system documentation
- ✓ Technical documentation

User Documentation

- ✓ Explain the features of the software and describe how to use them.
- ✓ It is intended to be read by the user of the software and is therefore expressed in the terminology of the application
  documentation
- ✓ User documentation is recognized as an important marketing tool
- ✓ Increases sales
- ✓ Many software developers hire technical writers to produce this part of their product

System Documentation

- ✓ To describe the software's internal composition so that the software can be maintained later in its life cycle.
- ✓ Commenting
- ✓ Indentation
- ✓ Naming Conventions

Technical Documentation

- ✓ to describe how a software system should be installed and serviced
- ✓ adjusting operating parameters
- ✓ installing updates
- ✓ reporting problems back to the software's developer

Summary

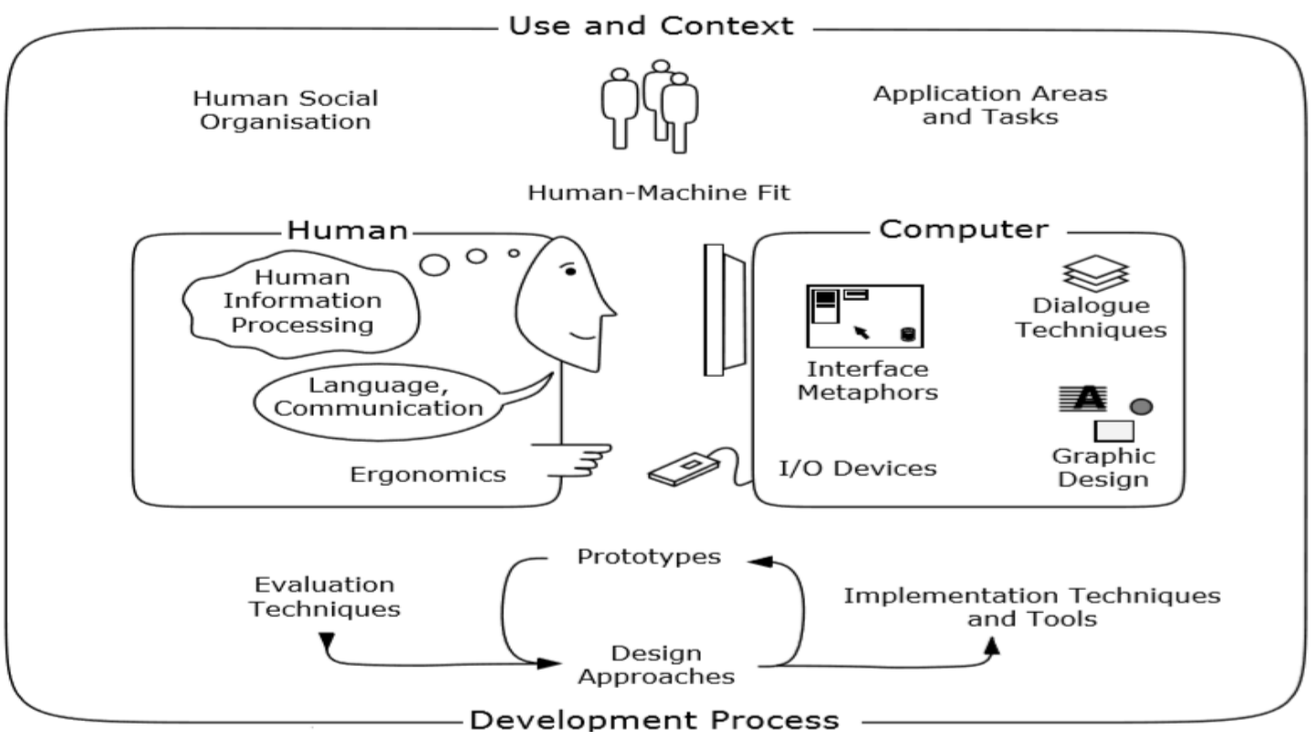
Documentation

- ✓ User Documentation
- ✓ System Documentation
- ✓ Technical Documentation

Topic-144. Human machine interface

Why its important?

- ✓ Presentation is everything for user!
- ✓ User is not interested to learn the inside of the software.



Ergonomics and Cognetics

- ✓ Ergonomics deals with designing systems that harmonize with the physical abilities of humans
- ✓ Cognetics deals with designing systems that harmonize with the mental abilities of humans

Ergonomics example

Human Machine Interface

- ✓ Why its important
- ✓ Ergonomics
- ✓ Cognetics

Topic-145. Software Ownership and Liability

Once software is made!

- ✓ One needs the ownership and should get profit from the investment.

Intellectual Property Law

- ✓ Legal Efforts to provide you ownership of the developed software.
- ✓ based on the well-established principles of:
  - ✓ Copyright
  - ✓ patent law

Purpose of Copyright or patent

- ✓ Allow the developer of a product to release that product to intended parties while protecting his or her ownership rights

Requirement

- ✓ Include statements in:
  - ✓ Requirement specifications
  - ✓ design documents
  - ✓ source code
  - ✓ test plans
  - ✓ in visible place within the final product

Software License

- ✓ Legal agreement between the owner and user of a software product that grants the user certain permissions to use the product without transferring ownership rights to the intellectual property.

Patent

- ✓ Patent laws were established to allow an inventor to benefit commercially from an invention.
- ✓ expensive and time-consuming to acquire
- ✓ Given the right to inventor for a limited period of time, which is typically 20 years

Consequences of breaking law

- ✓ In 2004, a little-known company, NPT Inc., successfully won a case against Research In Motion (RIM—the makers of the BlackBerry smartphones) for breaking the patent law few key technologies embedded in RIM's email systems

Consequences of breaking law

- ✓ The judgment included an injunction to suspend email services to all BlackBerry users in the United States! RIM eventually reached an agreement to pay NPT a total of $612.5 million, thereby averting a shutdown.

Summary

Software Ownership and Liability

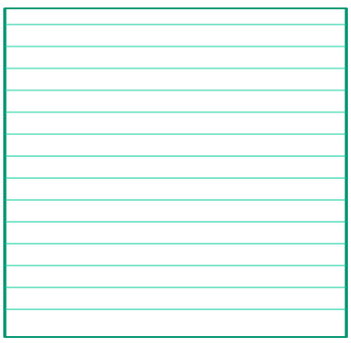
- ✓ Intellectual property law
- ✓ Copyright
- ✓ Patent
- ✓ Software License
- ✓ Consequence of breaking laws

Topic-146. Arrays and aggregates

Why we need data abstraction?

Arrays and Aggregates

Why we need data abstraction?



- ✓ User need to think as main memory organization
- ✓ Store family tree
- ✓ Store marks of 100 students

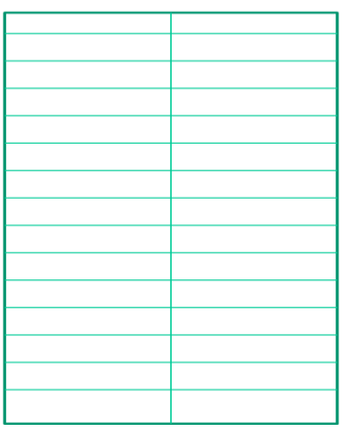
Arrays and Aggregates

Arrays

- ✓ Rectangular block of data whose entries are of the same type. Students.
- ✓ A one-dimensional array with 26 elements could be used to store the number of times each alphabet letter occurs in a page of text

Arrays and Aggregates

Two-dimensional Arrays



- ✓ A two-dimensional array consists of multiple rows and columns in which positions are identified by pairs of indices
- ✓ Colors of cars and their availability in the showroom

Aggregate type (Structures)

- ✓ Block of data items
  that might be of different types and sizes.
- ✓ The items within the block are usually
  called fields.

Name	
DOB	
Marks in Matric	
Address	
Semester	

Summary

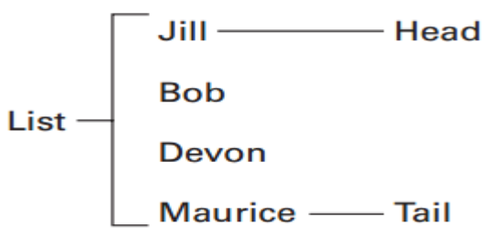
Arrays and aggregates

- ✓ Why we need data abstraction
- ✓ Arrays
- ✓ Aggregates types

Topic-147. List, stacks and queues

Lists

- ✓ A collection whose entries are arranged sequentially
- ✓ Almost any collection of data can be envisioned as a list,
- ✓ music recorded on a CD can be envisioned as a list of sounds.
- ✓ Guest lists, shopping lists, class enrollment lists, and
  inventory lists.
- ✓ By restricting the manner in which the entries of a list are accessed, we obtain two special types of lists known as stacks and queues



A list of names

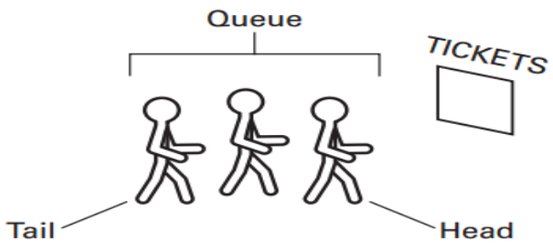
Stacks

- ✓ A stack is a list in which entries are inserted and removed only at the head
- ✓ Pushing: Inserting a new entry at the top of a stack.
- ✓ Popping: Removing an entry from the top of a stack.
- ✓ last-in, first-out (LIFO)
- ✓ **Top**: head of a stack is called the top of the stack
- ✓ **Bottom**: tail of a stack is called its bottom or base
- ✓ **Applications**: Recursion, Backtracking

Queue

- ✓ List in which the entries are removed only at the head and new entries are inserted only at the tail

- ✓ first-in, first-out (FIFO)
- ✓ Applications: underlying structure of a buffer

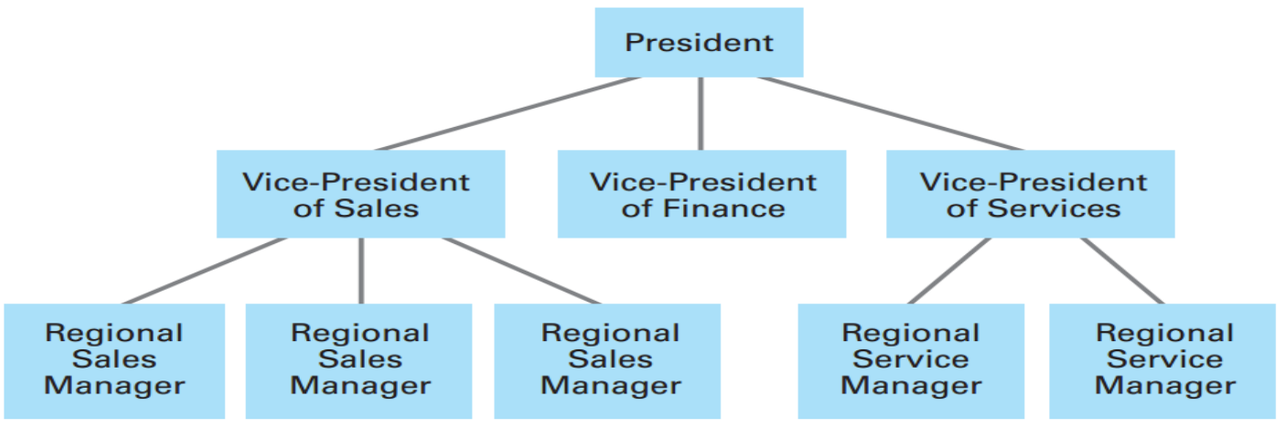


List, Stacks, and Queues

- ✓ What are lists
- ✓ How data is accessed in Stacks and Queues.

Topic-148. Trees

- ✓ A tree is a collection whose entries have a hierarchical organization
- ✓ **Constraint:** No individual in the company reports to two different superiors.



Node: Each position in a tree is called a node

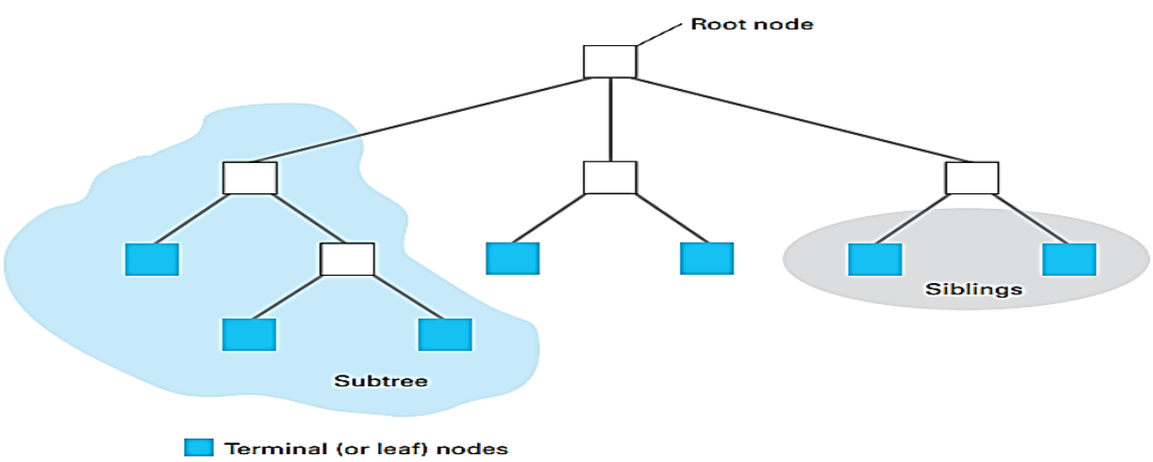
Root Node: The node at the top, base or root as in below figure.

Terminal Nodes: The nodes at the other extreme are called terminal nodes (or sometimes leaf nodes).
Depth: number of nodes in the longest path from the root to a leaf.

Children: immediate descendants
Parents: immediate ancestor
Sibling: nodes with the same parent.

Binary Tree: A tree in which each parent has no more than two children

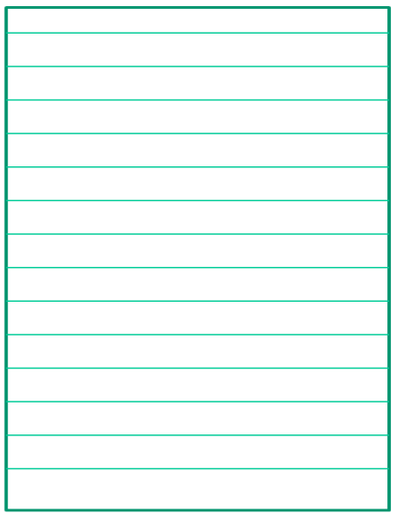
Subtrees: n is called a binary tree. If we select any node in a tree, we find that that node together with the nodes below it also have the structure of a tree. We call these smaller structures
subtrees
Branch: Each such subtree is called a branch from the parent.



Trees

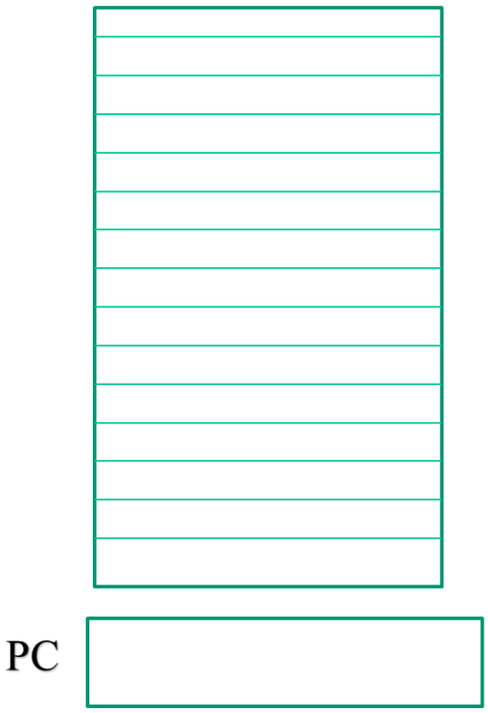
- ✓ Why we need tree data structure?
- ✓ Basic terminologies

Pointers



- ✓ A pointer is a storage area that contains encoded address of memory cells.
- ✓ For example, if we must repeatedly move an item of data from one location to another, lets designate the fixed location to serve as pointer. Each time, item is moved, the pointer is updated.

Pointers

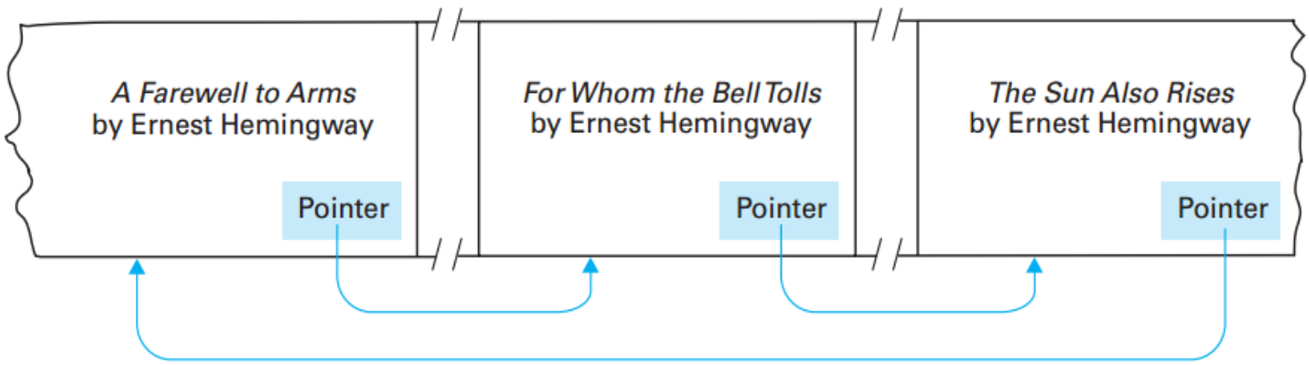


Program Counter

- ✓ Program counter is used to hold the address of the next instruction to be executed. Thus, the program counter plays the role of a pointer. In fact, another name for a program counter is instruction pointer.

Another Application

- ✓ Suppose we want to link all novels by one author, lets add a pointer as additional memory cell.
- ✓ Novels with common authorship can be linked



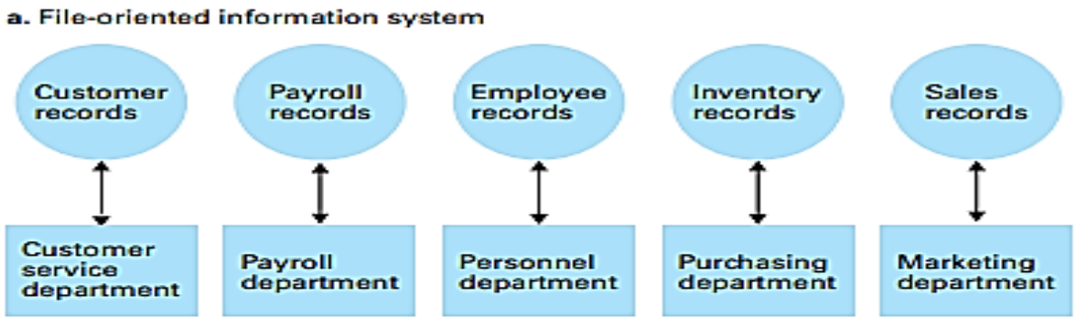
Pointers

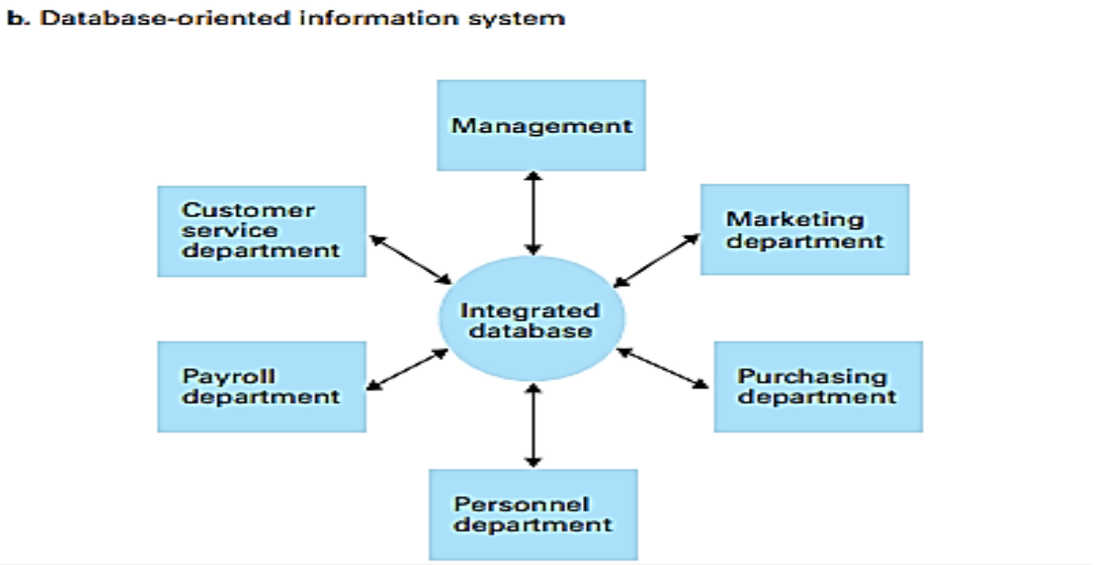
- ✓ Why pointer
- ✓ PC – Instruction pointer
- ✓ Applications of pointers

Topic-150. The Significance of Database Systems

Flat File

- ✓ one-dimensional storage system, meaning that it presents its information from a single point of view
- ✓ Addition
- ✓ Edit
- ✓ Deletion
- ✓ Querying





Advantages of DB

- ✓ Helps in making informed decisions especially when combined with data mining technologies.
- ✓ Google, eBay, Amazon, Nadra, FBR etc provide interface between clients and databases

The Significance of Database Systems

- ✓ Database
- ✓ Flat file
- ✓ Advantages of DB

Topic-151. Database Systems: Role of Schema - I

Give only restricted information

- ✓ Someone placing an order should not have access to the company's financial data.
- ✓ Employees' supervisors can ask for the information of employees records, but should not be given access to corporate inventory or sales record.
- ✓ Implemented as Schema and subschema.

Schema and Subschema

- ✓ A schema is a description of the entire database structure that is used by the database software to maintain the database
- ✓ A subschema is a description of only that portion of the database pertinent to a particular user's needs.

Example

- ✓ Schema of university database include student complete record, courses registered by students, student advisor information, Faculty members employment history etc.
- ✓ Registrar should not be able to see the employment history, but can see student advisor

Example continued

- ✓ The subschema for the payroll department would provide the employment history of each faculty member but would not include the linkage between students and advisers.
- ✓ Examination department should not be able to access salary information.

Summary

Role of Schema

- ✓ Restricted Information Accessibility
- ✓ Subschema
- ✓ Schema

Topic-152. Database Systems: Role of Schema - II

Give only restricted information

- ✓ Someone placing an order should not have access to the company's financial data.
- ✓ Employees' supervisors can ask for the information of employees records, but should not be given access to corporate inventory or sales record.
- ✓ Implemented as Schema and subschema.

Schema and Subschema

- ✓ A schema is a description of the entire database structure that is used by the database software to maintain the database
- ✓ A subschema is a description of only that portion of the database pertinent to a particular user's needs.

Example

- ✓ Schema of university database include student complete record, courses registered by students, student advisor information, Faculty members employment history etc.
- ✓ Registrar should not be able to see the employment history, but can see student advisor

Example continued

- ✓ The subschema for the payroll department would provide the employment history of each faculty member but would not include the linkage between students and advisers.
- ✓ Examination department should not be able to access salary information.

Summary

Role of Schema

- ✓ Restricted Information Accessibility
- ✓ Schema
- ✓ Subschema

Topic-153. Database Systems: Relational Database Model

Database Model

- ✓ Abstraction tool – Hides the internal complexities of DB storage and presents the conceptual view of the data.
- ✓ DBMS has then commands to translate the conceptual view into the actual database.
- ✓ This conceptual view is called Database Mod

Relational Database Model

- ✓ The conceptual view of the database is that of a collection of tables consisting of rows and columns.
- ✓ It portrays data as being stored in rectangular tables, called relations.
- ✓ A row in a relation is called a tuple

Relational Database Model

- ✓ Columns in a relation are referred to as attributes
- ✓ The information is not actually stored in rows and columns

Emp-No	Name	F-Name	Address	Salary	department	Date-of-Joining

Summary

Relational Database Model

- ✓ Database Model
- ✓ Relational Database Model

Topic-154. Database Systems: Issues of Relational Designs

Scenario

Empl Id	Name	Address	SSN
25X15 34Y70 23Y34 • • •	Joe E. Baker Cheryl H. Clark G. Jerry Smith • • •	33 Nowhere St. 563 Downtown Ave. 1555 Circle Dr. • • •	111223333 999009999 111005555 • • •

We want to include job history associated with each employee that consists of such attributes as job title (secretary, office manager, floor supervisor), a job identification code (unique to each job), the department in which the job

exists, and the period during which the employee held the job in terms of a starting date and termination date. (We use an asterisk as the termination date if the job represents the employee's current position.)

Scenario

Empl Id	Name	Address	SSN	Job Id	JobTitle	Skill Code	Dept	Start Date	Term Date
25X15	Joe E. Baker	33 Nowhere St.	111223333	F5	Floor manager	FM3	Sales	9-1-2009	9-30-2010
25X15	Joe E. Baker	33 Nowhere St.	111223333	D7	Dept. head	K2	Sales	10-1-2010	*
34Y70	Cheryl H. Clark	563 Downtown Ave.	999009999	F5	Floor manager	FM3	Sales	10-1-2009	*
23Y34	G. Jerry Smith	1555 Circle Dr.	111005555	S25X	Secretary	T5	Personnel	3-1-1999	4-30-2010
23Y34	G. Jerry Smith	1555 Circle Dr.	111005555	S26Z	Secretary	T6	Accounting	5-1-2010	*
•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•

- ✓ Issues: redundancy, when to delete employee, when to delete jobs
- ✓ Issues Resolved?
- ✓ Redundancy
- ✓ When to add, delete employee,
- ✓ When to add, delete jobs

Empl Id	Name	Address	SSN
25X15 34Y70 23Y34	Joe E. Baker Cheryl H. Clark G. Jerry Smith	33 Nowhere St. 563 Downtown Ave. 1555 Circle Dr.	111223333 999009999 111005555

JOB relation

Job Id	JobTitle	Skill Code	Dept
S25X S26Z F5 • • •	Secretary Secretary Floor manager • • •	T5 T6 FM3 • • •	Personnel Accounting Sales • • •

ASSIGNMENT relation

Empl Id	Job Id	Start Date	Term Date
23Y34 34Y70 23Y34 • • •	S25X F5 S26Z • • •	3-1-1999 10-1-2009 5-1-2010 • • •	4-30-2010 * * • • •

Summary

Issues of Relational Design

- ✓ Scenario helped to learn the issues
- ✓ The raised issues were solved by putting only relevant information in each table

Topic-155. Database Systems: Relational Operators

Select Operation

Empl Id	Name	Address	SSN
25X15	Joe E. Baker	33 Nowhere St.	111223333
34Y70	Cheryl H. Clark	563 Downtown Ave.	999009999
23Y34	G. Jerry Smith	1555 Circle Dr.	111005555
•	•	•	•
•	•	•	•
•	•	•	•

- ✓ Now, we are ready to extract the required information.
- ✓ Select: Gives you the required tuple/s, the result of such a process is another relation consisting of the tuples selected from the parent relation.

Project Operation

Empl Id	Name	Address	SSN
25X15	Joe E. Baker	33 Nowhere St.	111223333
34Y70	Cheryl H. Clark	563 Downtown Ave.	999009999
23Y34	G. Jerry Smith	1555 Circle Dr.	111005555
•	•	•	•
•	•	•	•
•	•	•	•

- ✓ Project: The PROJECT operation extracts columns.
- ✓ The JOIN of two relations produces a new relation whose attributes consist of the attributes from the original relations.

Empl Id	Name	Address	SSN
25X15	Joe E. Baker	33 Nowhere St.	111223333
34Y70	Cheryl H. Clark	563 Downtown Ave.	999009999
23Y34	G. Jerry Smith	1555 Circle Dr.	111005555

JOB relation

Job Id	Job Title	Skill Code	Dept
S25X	Secretary	T5	Personnel
S26Z	Secretary	T6	Accounting
F5	Floor manager	FM3	Sales
•	•	•	•
•	•	•	•
•	•	•	•

ASSIGNMENT relation

Empl Id	Job Id	Start Date	Term Date
23Y34	S25X	3-1-1999	4-30-2010
34Y70	F5	10-1-2009	*
23Y34	S26Z	5-1-2010	*
•	•	•	•
•	•	•	•
•	•	•	•

Summary

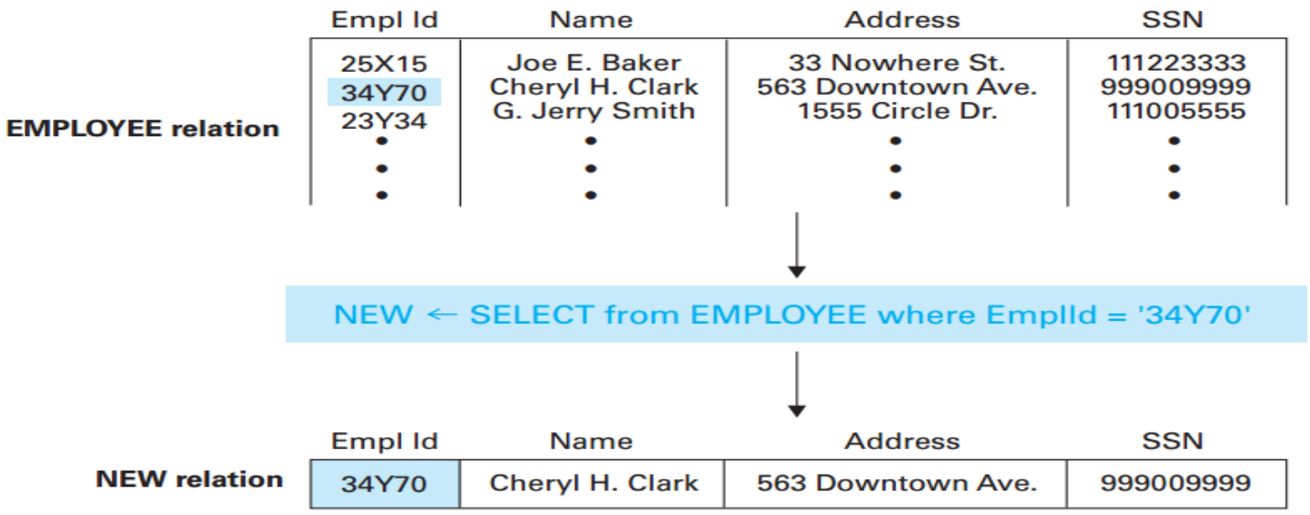
Relational Operators

- ✓ Extraction of Data
- ✓ Select Operation
- ✓ Project Operation
- ✓ Join Operation

Topic-156. Database Systems: Select Operation

Select: Gives you the required tuple/s, the result of such a process is another relation consisting of the tuples selected from the parent relation.

NEW ← SELECT from EMPLOYEE where EmpId = '34Y70'



Emp-id	Name	Salary
1001	Ahmad	50000
1002	Ali	60000
1003	Ayan	80000
1004	Akbar	70000

NEW ← SELECT from EMPLOYEE where Salary >= 65000

Emp-id	Name	Salary
1003	Ayan	80000
1004	Akbar	70000

Summary

Select Operation

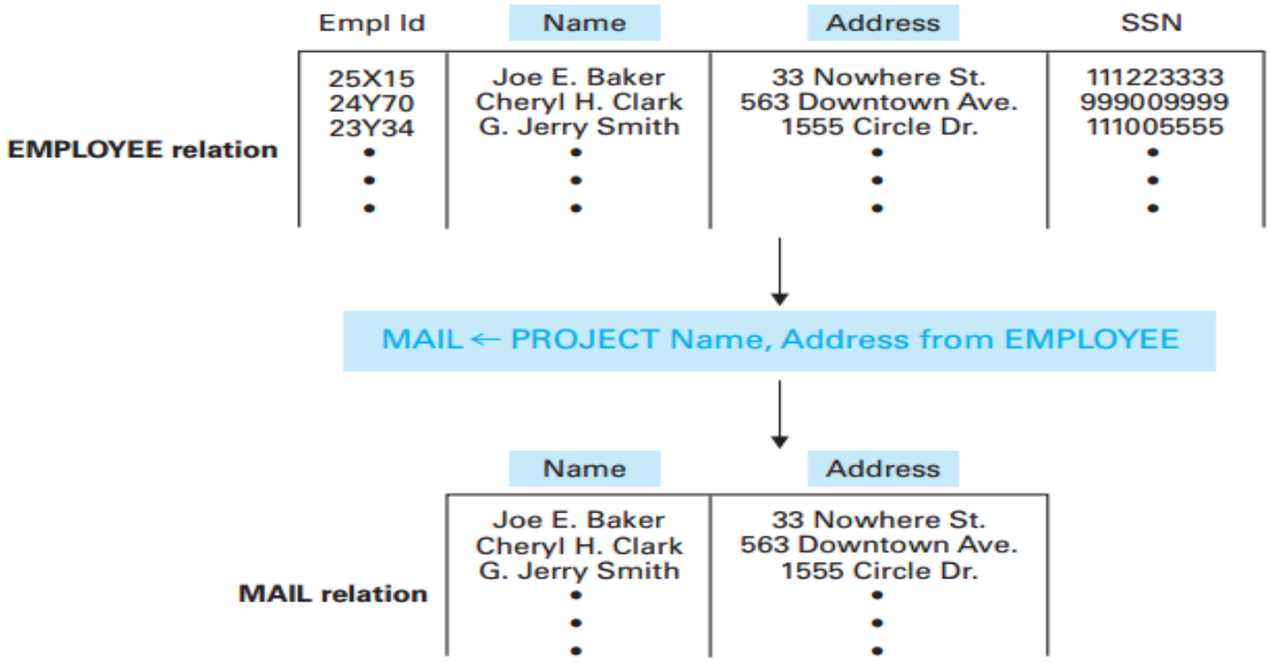
- ✓ Selection based on some characteristics
- ✓ Examples

Topic-157. Database Systems: Project Operation

Project Operation

Project: The PROJECT operation extracts the required columns.

Example:



Emp-id	Name	Salary
1001	Ahmad	50000
1002	Ali	60000
1003	Ayan	80000
1004	Akbar	70000

NEW ← Project Name, Salary from EMPLOYEE

Name	Salary
Ahmad	50000
Ali	60000
Ayan	80000
Akbar	70000

Project Operation

- ✓ Projection from existing relation

examples

Topic-158. Database Systems: Join Operation

- ✓ The JOIN of two relations produces a new relation whose attributes consist of the attributes from the original relations

Registration no	Name	Father Name
VU201912	Ali	Asghar
VU201913	Ahmad	Bilal

Registration no	Marks in ITC
VU201912	95
VU201913	80

Registration no	Name	Father Name	Marks in ITC
VU201912	Ali	Asghar	95
VU201913	Ahmad	Bilal	80

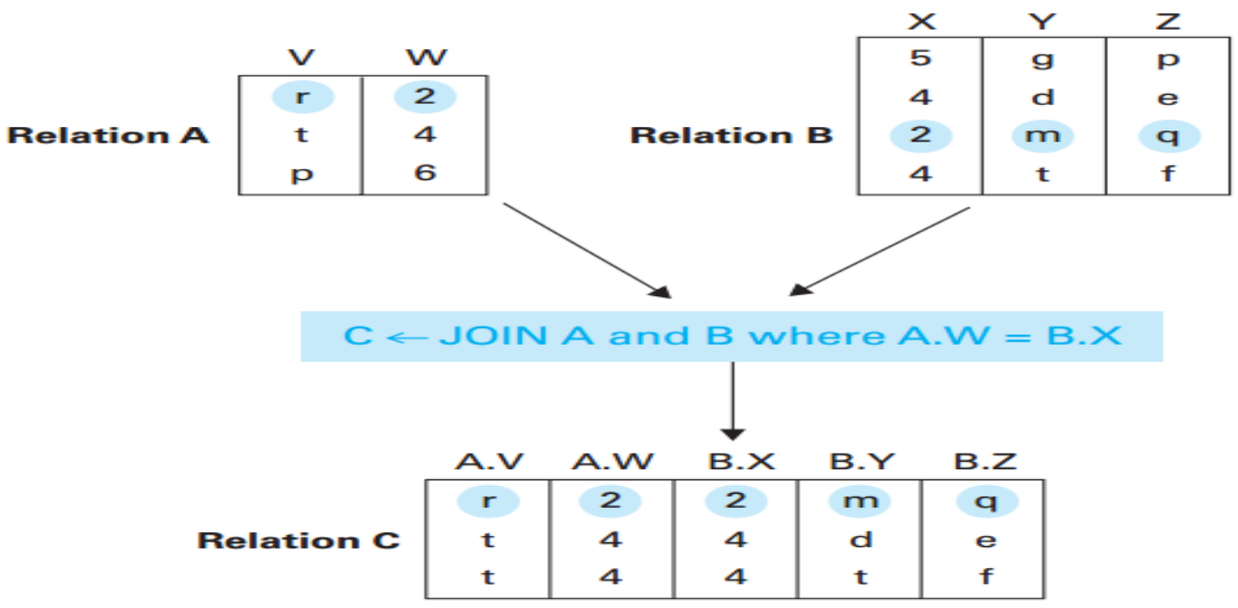
Attributes Names after Joining

- ✓ Relation A containing attributes V and W is joined with relation B containing attributes X, Y, and Z, then the result has five attributes named A.V, A.W, B.X, B.Y, and B.Z.) This naming convention ensures that the attributes in the new relation have unique names, even originals have same names

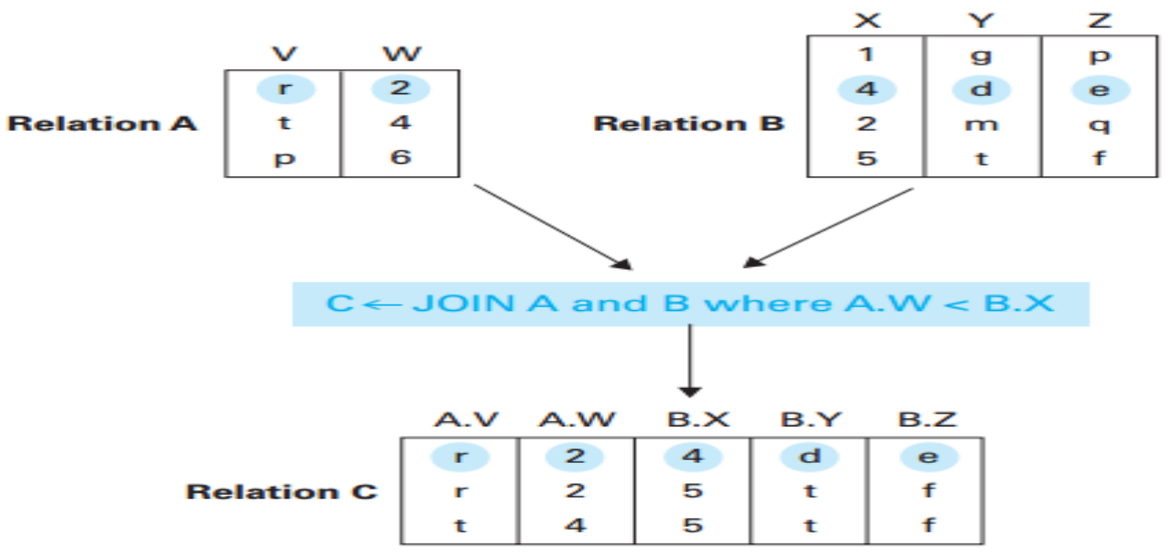
Tuples after Joining

- ✓ The tuples (rows) of the new relation are produced by concatenating tuples from the two original relations. Which tuples are actually joined to form tuples in the new relation is determined by the condition under which the JOIN is constructed.

Example 1



Example 2



Summary

Join Operation

- ✓ What happens when two relations are joined in terms of Attributes and Data
- ✓ Discussed two examples

Topic-159. Database Systems: Object Oriented Databases

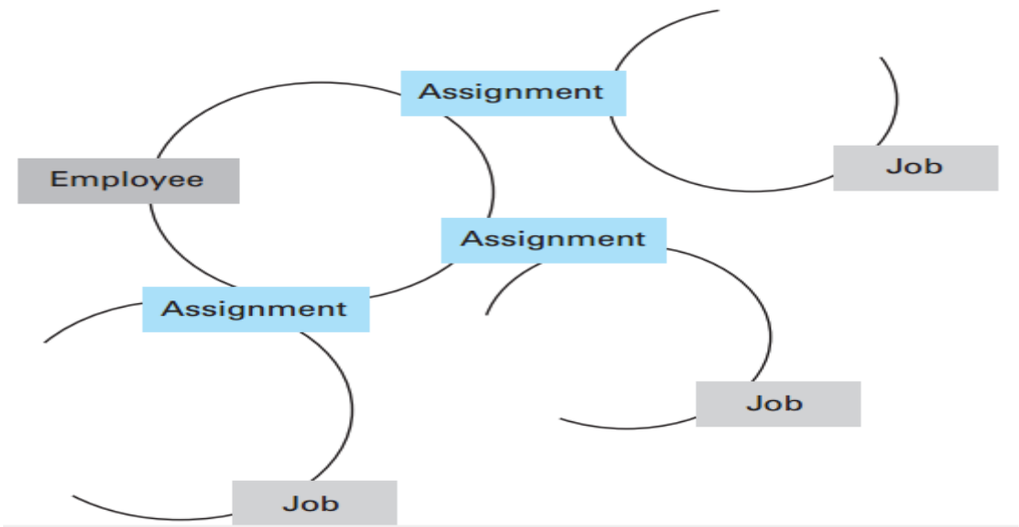
- ✓ Database model is based on the object-oriented paradigm.
- ✓ Database consisting of objects that are linked to each other to reflect their relationships.

Lets map the Employee example

- ✓ could consist of three classes (types of objects): Employee, Job, and Assignment.
- ✓ Employee object: contains EmplId, Name, Address, and SSNum;
- ✓ Job Object: JobId, JobTitle, SkillCode, and Dept;
- ✓ Assignment Object: StartDate and TermDate

Lets map the Employee example

- ✓ Employee to Assignment
- ✓ Assignment to Job



Links management

- ✓ The links between objects in an object-oriented database are normally maintained by the DBMS.
- ✓ In an application software, programmer just specify which object should be linked with whom?
- ✓ DBMS might link using lists we studied in last modules.

Transient Objects

- ✓ Normally, when an object-oriented program is executed, the objects created during the program's execution are discarded when the program terminates. In this sense the objects are considered transient.

Persistent Objects

- ✓ Objects that are created and added to a database must be saved after the program that created them terminates. Such objects are said to be persistent.

Object-oriented DB VS relational DB

- ✓ In Object-oriented DB, entire software system (application software, DBMS, and the database itself) to be designed in the same paradigm.
- ✓ In imperative paradigm, relational DB is queried, making it non-homogenous.

Object-Oriented Databases

- ✓ Employee Example

- ✓ Link Management
- ✓ Transient Object
- ✓ Persistent Object
- ✓ Object-Oriented DB vs Relational DB

Topic-160. Database Systems: Maintaining DB Integrity

Inexpensive DBMS

- ✓ Generally for personal use.
- ✓ Have a single objective—To shield the user from the technical details of the database implementation.
- ✓ Databases maintained by these systems are relatively small and generally contain information whose loss or corruption would be inconvenient rather than disastrous.

Inexpensive DBMS: What happens when problem arises

- ✓ The user can usually correct the erroneous items directly or reload the database from a backup copy and manually make the modifications required to bring that copy up to date.

Commercial DB Systems

- ✓ Normally this DB is large, multiuser, and commercial. The cost of incorrect or lost data can be enormous and can have devastating consequences.

Commercial DB Systems – Major Role

- ✓ To maintain the database's integrity by guarding against problems such as operations that for some reason are only partially completed or causing inaccurate information in the database

Commercial DB Systems – Major Role

- ✓ The Commit/Rollback Protocol
- ✓ Locking

Summary

Maintaining DB Integrity

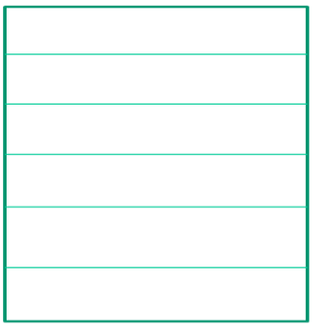
- ✓ Personal DBMS
- ✓ Commercial DBMS
- ✓ Major Roles of Commercial DBMS

Topic-161. Database Systems: The Commit/Rollback Protocol

Single transaction having multiple steps

- ✓ The transfer of funds from one bank account to another
- ✓ The cancellation of an airline reservation
- ✓ The registration of a student in a university course

The Commit/Rollback Protocol



Transfer of bank account

✓ One account needs to be decremented, other account needs to be incremented. In between the information is inconsistent.

How to handle such situation

- ✓ By maintaining a log containing a record of each transaction's activities in a non volatile storage system.
- ✓ Before a transaction alters the database, log file is updated.

Commit Point

- ✓ The point at which all the steps in a transaction have been recorded in the log is called the commit point.

Roll back (Undo)

- ✓ If problems should arise before a transaction has reached its commit point, the DBMS might find itself with a partially executed transaction that cannot be completed. In this case the log can be used to roll back (undo) the activities.

Roll back Applications

- ✓ A transaction might be terminated before it has completed all its steps because of an attempt to access privileged information.
- ✓ Can be used in a deadlock in which competing transactions find themselves waiting for data being used by each other.

Cascading Rollback

- ✓ Suppose a transaction being rolled back have updated an account balance, and another transaction have already used the updated value, such all transactions need to be rolled back. The result is the problem known as cascading rollback.

Summary

The Commit/Rollback Protocol

- ✓ Transactions involving Multiple Steps
- ✓ Commit Log
- ✓ Roll Back
- ✓ Cascading Rollback

Topic-162. Database Systems: Locking

Example : Incorrect Summary Problem

- ✓ If one transaction is in the middle of transferring funds from one account to another when another transaction tries to compute the total deposits in the bank.

Example : Lost Update problem

- ✓ Suppose two transactions want to deduct from the same account. If one reads the account's current balance at the point when the other has just read the balance but has not yet updated the new balance, then both transactions will base their deductions on the same initial balance.

One solution

- ✓ DBMS could force such transactions to execute entirely.
- ✓ However, mass storage operations take too much time and in time sharing paradigm, it would be time consuming, at least other transaction which does not want access of this critical data, should be given chance of execution.

Locking protocol

- ✓ 2nd solution is to lock that particular item in the memory, no one can use it unless it is unlocked by the parent transaction.

Locks types

- ✓ Shared locks
- ✓ Exclusive locks

Shared Locks

- ✓ If the transaction is not going to change the data, then one can use shared lock, this means other transactions can view the record, but can not update the value.

Exclusive lock

- ✓ If the transaction is going to alter the item, it must have exclusive access, meaning that it must be the only transaction with access to that data.

Summary

Locking

- ✓ Incorrect Summary Problem
- ✓ Lost Update Problem
- ✓ Locking
- ✓ Shared Locks
- ✓ Exclusive Locks

Topic-163. Traditional File Structures: Sequential Files

Why study Traditional File Structures

- ✓ Database has emerged from the historical structures of file.
- ✓ Technologies like indexing and hashing are important tools in construction of massive DBs.

Definition

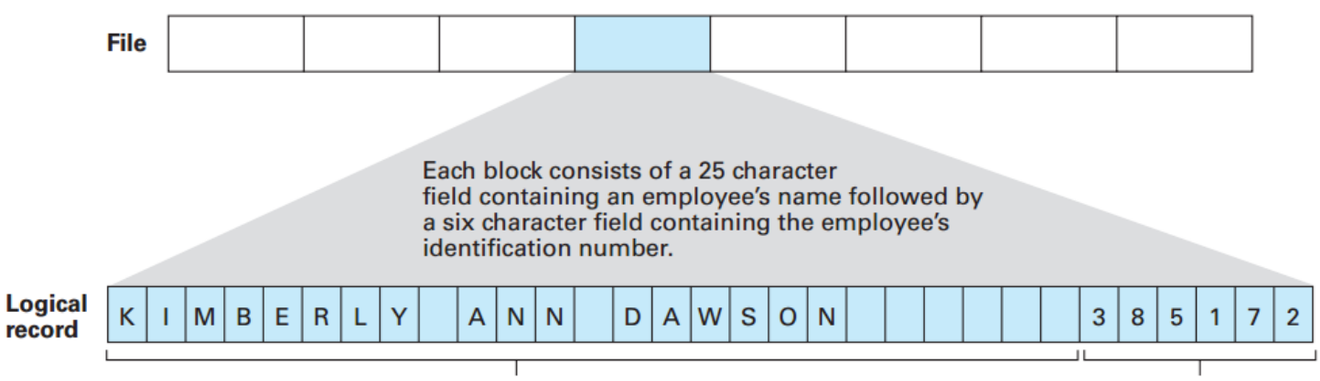
- ✓ A sequential file is a file that is accessed in a serial manner from its beginning to its end as though the information in the file were arranged in one long row.

Applications

- ✓ Examples include audio files, video files, files containing programs, and files containing textual documents.
- ✓ Most of the files created by a typical personal computer user are sequential files.
- ✓ Excel file is stored as a sequential file at the backend.

Example

- ✓ Storing employees record, lets store name in 25 character, and employee identification number in 6 characters = total 31 characters



Data storage in the drives

- ✓ Data should be stored as sequence.
- ✓ In case of magnetic tape or CD, its straight forward as they store data in a sequence.
- ✓ However, in the case of magnetic disk storage, file would be scattered in different sectors. Sector information is stored as a file in the disk's directory system on the same disk.

End-of-file (EOF)

- ✓ In processing a sequential file is the need to detect when the end of the file is reached. Generically, we refer to the end of a sequential file as the end-of-file (EOF). as a file in the disk's directory system on the same disk.
- ✓ One way is to add a special record Sentinel at the end of the file.

- ✓ Another way is to place this information in the system directory file.

```
while (the EOF has not been reached):
  retrieve the next record from the file and process it
```

Summary

Sequential Files

- ✓ Traditional File Structures
- ✓ Sequential Files
- ✓ Applications
- ✓ Storage
- ✓ EOF

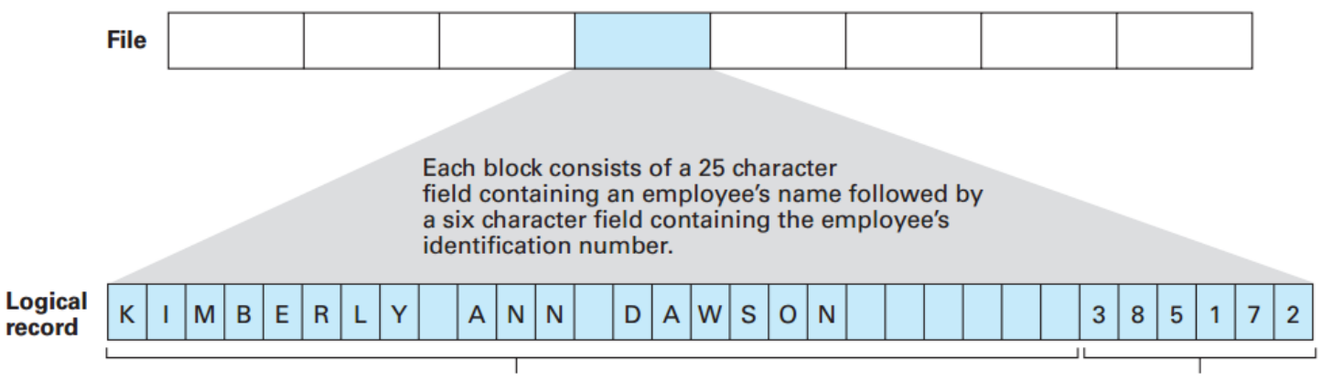
Topic-164. Traditional File Structures: Indexed Files

Why we need Indexed Files?

- ✓ When information needs to be retrieved in a sequence, sequential files fine, however, when one wants to retrieve in unpredictable manner what can be done?

Example

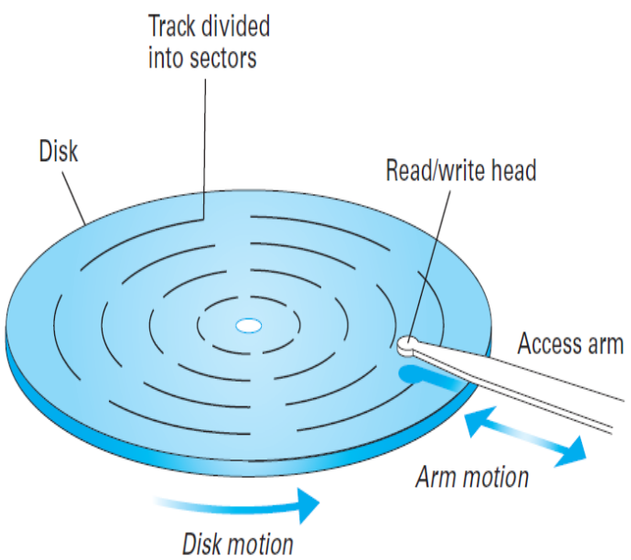
- ✓ Retrieve employee number 6



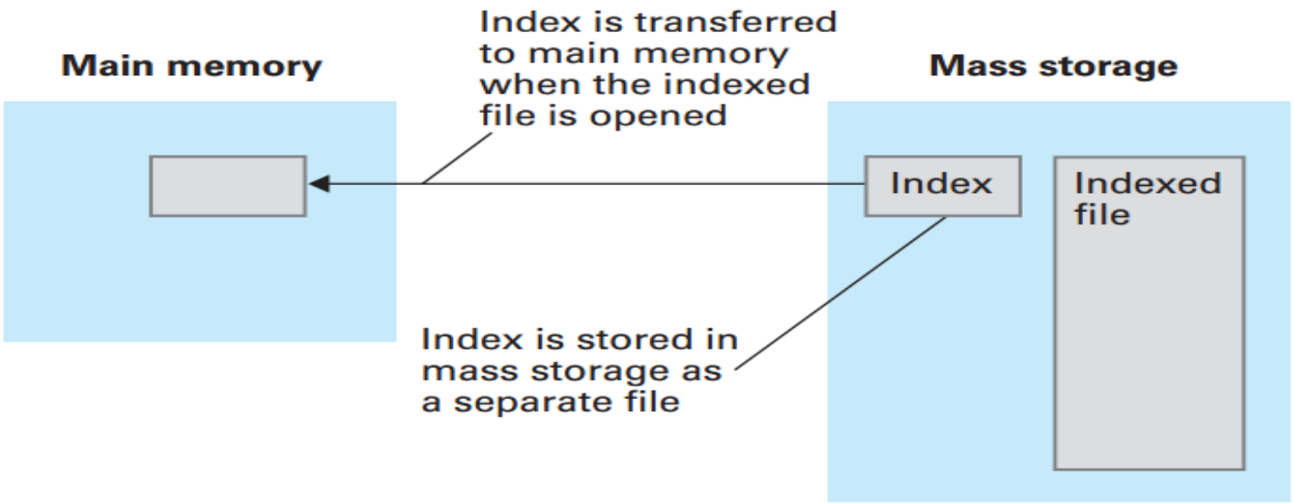
Example

- ✓ Retrieve employee number 6

Employee Number	Address
1	AA001101ED
2	BA011001EE
3	CC001101DE
4	00ABDE01ED
5	DCEB001010
6	CCB01101ED
7	11001101ED



Index and Indexed Files Storage



Summary

Indexed Files

- ✓ Why we need Indexed Files?
- ✓ Example
- ✓ Index Storage and Retrieval

Topic-165. Traditional File Structures: Hash Files

Disadvantage of Index files

- ✓ Index needs to be maintained in the mass storage, needs to be read and put in the main memory before reading data from the file.

Hashing

- ✓ Technique that provides similar access without storing index as overhead.
- ✓ Instead of looking into the index, hashing identifies the location of the record directly from the key.

Hash system

- ✓ The data storage space is divided into several sections, called buckets, each of which is capable of holding several records. directly from the key.

Hash function

- ✓ The records are dispersed among the buckets according to an algorithm that converts key values into bucket numbers. (This conversion from key values to bucket numbers is called a hash function).

Hash file VS Hash function

- ✓ When hashing is applied to a storage structure in mass storage, the result is called a hash file.
- ✓ When applied to a storage structure within main memory, the result is usually called a hash table.

Summary

Hash Files

- ✓ Why we need Hashing?
- ✓ Hash Functions, Hash Files, Hash Tables

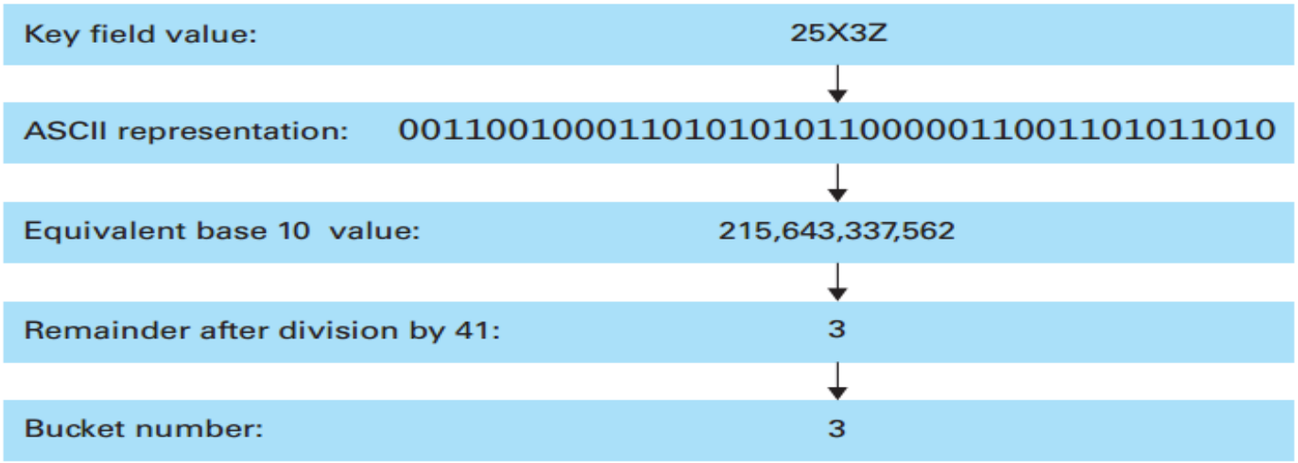
Topic-166. Traditional File Structures: Hash Files Examples

Hash Function

- ✓ Hash function is the key factor to store and retrieve data.
- ✓ Hash function can be made using variety of ways.
- ✓ One simple method is remainder method.

Example – Hash Function (Remainder)

- ✓ Lets suppose we have 41 buckets numbered from 0 to 40. Let's find bucket number for employee 25X3Z.

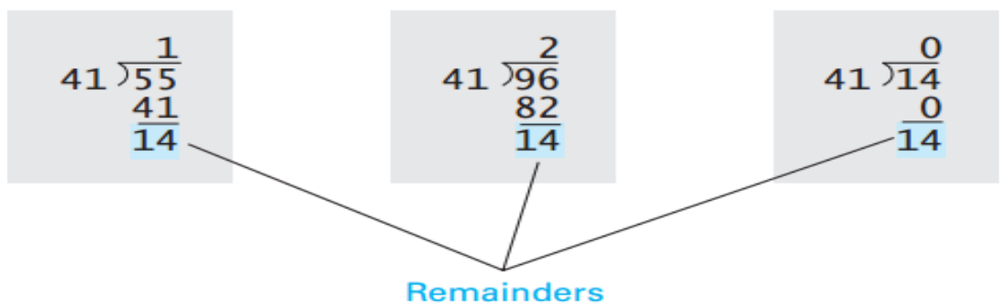


Clustering

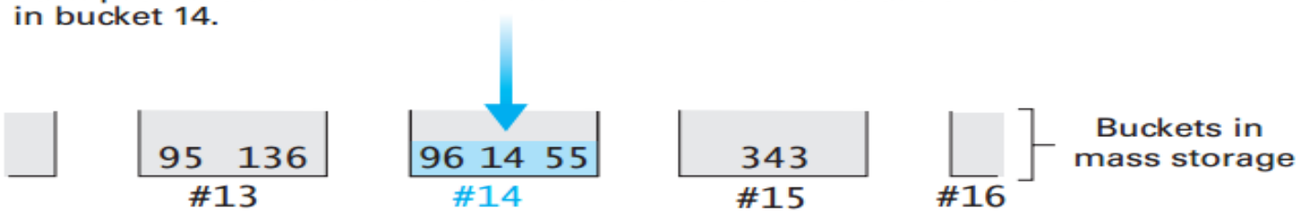
- ✓ **Goal of Hashing**: The records being stored should be distributed evenly among the buckets.
  Clustering: Disproportionate number of keys happen to hash to the same bucket.

Example

- ✓ Employee number going to be stored in buckets



When divided by 41, the key field values of 14, 55, and 96 each produce a remainder of 14. Thus these records are stored in bucket 14.



Summary

Hash Files Examples

- ✓ Hash Function
- ✓ Examples
- ✓ Remainder Method
- ✓ Clustering

Topic-167. Database Systems: Data Mining

- ✓ A rapidly expanding subject that is closely associated with database technology is data mining, which consists of techniques for discovering patterns in collections of data.

Application Areas

- ✓ Marketing
- ✓ Inventory management

- ✓ Quality control
- ✓ Loan risk management
- ✓ Fraud detection
- ✓ Investment analysis of data
- ✓ Identifying the functions of particular genes encoded in DNA molecules

Data Mining Focus

- ✓ To identify previously unknown patterns as opposed to traditional database inquiries that merely ask for the retrieval of stored facts.

Data Mining Practice

- ✓ Data mining is practiced on static data collections, called data warehouses, rather than “online” operational databases that are subject to frequent updates.

Common Forms of DM

- ✓ Class description
- ✓ Class discrimination

Class Description

- ✓ Deals with identifying properties that characterize a given group of data items.
- ✓ E.g. to identify characteristics of people who buy small economical vehicles.

Class Discrimination

- ✓ Identifying properties that divide two groups, e.g. to find properties that distinguish customers who shop for used cars from those who shop for new ones.

Summary

Data Mining

- ✓ Why to perform Data Mining?
- ✓ Application Areas
- ✓ Focus and Practice
- ✓ Class Description
- ✓ Class Discrimination

Topic-168. Database Systems: Data Mining Examples and Implications

Cluster Analysis

- ✓ Seeks to discover classes, cluster analysis tries to find properties of data items that lead to the discovery of groupings.

Cluster Analysis Example

- ✓ In analyzing information about people’s ages who have viewed a particular motion picture, cluster analysis might find that the customer base breaks down into two age groups—a 4-to-0 age group and a 25-to-40 age.

Association Analysis

- ✓ Involves looking for links between data groups, e.g. it is association analysis that might reveal that customers who buy potato chips also buy cold drink.

Outlier Analysis

- ✓ Tries to identify data entries that do not comply with the norm.
- ✓ Outlier analysis can be used to identify errors in data collections
- ✓ To identify credit card theft by detecting sudden deviations from a customer’s normal purchase patterns

Sequential Pattern Analysis

- ✓ Tries to identify patterns of behavior over time.

- ✓ E.g. sequential pattern analysis might reveal trends in environmental systems such as climate conditions.

Summary

Data Mining Examples and Implications

- ✓ Cluster Analysis
- ✓ Association Analysis
- ✓ Outlier Analysis
- ✓ Sequential Pattern Analysis

Topic-169. Database Systems: Social Impact of Database Technology

Database Technology

- ✓ Database Technology has affected our lives in many aspects:
- ✓ Automated library systems place a patron's reading habits within easy reach.
- ✓ Internet search engines keep records of their clients' requests.

Data Collection

- ✓ Volunteers fill the surveys, or government agencies make a regulations when you need to give information.
- ✓ However, with DB technology, data is being automatically retrieved.
- ✓ Grocery store asks customers to fill forms to offer discounts.

Automatic Data Collection:

- ✓ Credit company records the purchasing practices.
- ✓ Websites record who visits the site.
- ✓ Social activists who record the license plate numbers on the cars parked in a targeted institution's parking lot.

Data is Currency

- ✓ Targeted marketing
- ✓ Data is linked
- ✓ Subscription forms for body-building magazines can be mailed to those who have recently purchased exercise equipment
- ✓ Dog obedience magazines can be targeted toward those who have recently purchased dog food

Data Protections

- ✓ Several approaches to protecting society from abusive use of data.
- ✓ Legal remedies
- ✓ Passing a law is not enough. It does not stop the action, it merely makes it illegal.
- ✓ Sharing of contact information.
- ✓ Public Opinion: In the early 1990s it was public opinion that ultimately stopped major credit bureaus from selling mailing lists for marketing purposes.
- ✓ Google discontinued its Google Buzz social networking tool in 2011 after public criticism on automatic sharing of contact information.

Summary

Social Impact of Database Technology

- ✓ Database Technology
- ✓ Data Collection and Storage

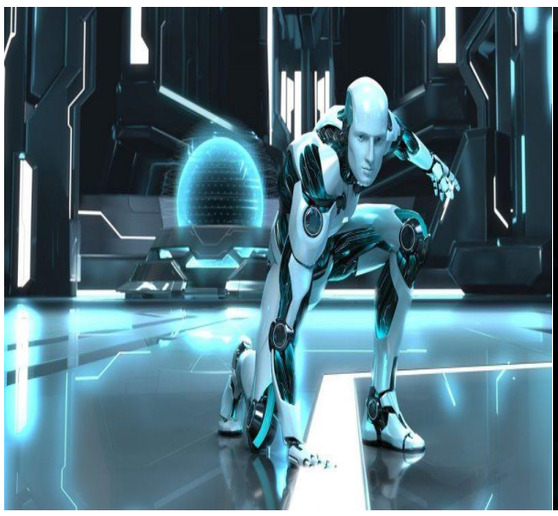
- ✓ Data Protection

Topic-170. Artificial Intelligence: Introduction and Vision

- ✓ Artificial intelligence is the field of computer science that seeks to build autonomous machines - machines that can carry out complex tasks without human intervention

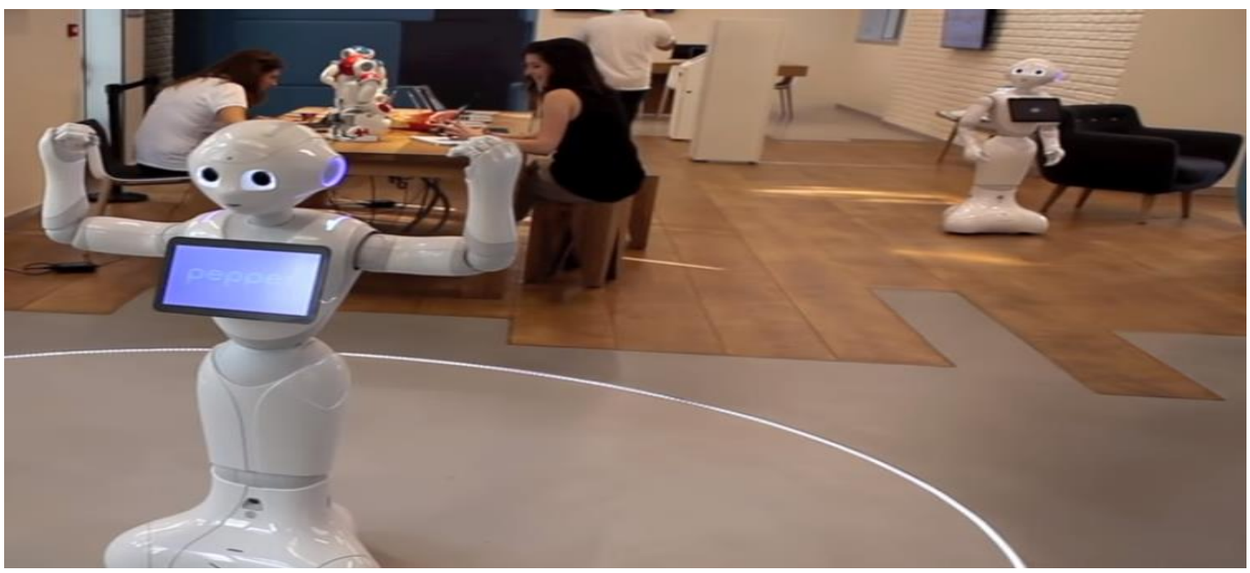
Applications

- ✓ Robots
- ✓ Self Driving Cars

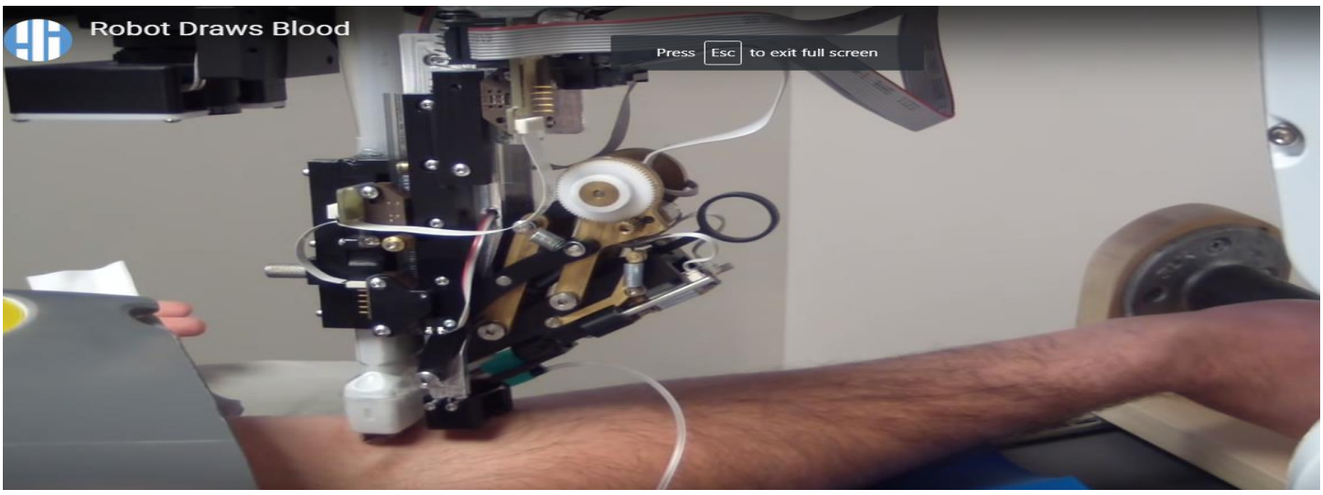



Medical Robot can alleviate patients even 40 times a day!

Pepper Robot works as receptionist in Belgian hospital



- ✓ Image analysis to find veins
- ✓ Confirmation by ultrasound
- ✓ Accurate than human



Applications

- ✓ TUG robot can carry 400KG of medication
- ✓ Millimeter sized Microbot delivers drugs through bloodstream at targeted area.

Applications

- ✓ Cyber security defense
- ✓ Recruiting automation
- ✓ Intelligent conversational interfaces
- ✓ Reduced energy used and cost
- ✓ Predicting vulnerability
- ✓ Market prediction
- ✓ Accelerated reading

AI as field

- ✓ Merges with other subjects such as psychology, neurology, mathematics, linguistics, and electrical and mechanical engineering targeted area.

Summary

AI: Introduction and Vision

- ✓ Robots
- ✓ Self driving cars
- ✓ More applications
- ✓ AI as a field

Topic-171. Artificial Intelligence: Intelligent Agents

Agent

- ✓ Much of the research in artificial intelligence can be categorized in terms of an agent's behavior.
- ✓ An agent is a "device" that responds to stimuli from its environment. It is natural to envision an agent as an individual machine such as a robot.

Agent Examples

- ✓ Agent may take other forms such as an autonomous airplane, a character in an interactive video game, or a process communicating with other processes over the Internet (perhaps as a client, a server, or a peer)

Data Acquisition – Agent

- ✓ Most agents have sensors by which they receive data from their environments and actuators by which they can affect their environments.

Sensor Examples

- ✓ Microphones
- ✓ Cameras

- ✓ Range sensors
- ✓ Air or soil sampling devices

Actuators Examples

- ✓ Wheels
- ✓ Legs
- ✓ Wings
- ✓ Grippers
- ✓ Speech synthesizers

AI Research

- ✓ Can be characterized in the context of building agents that behave intelligently, meaning that the actions of the agent's actuators must be rational responses to the data received though its sensors.
- ✓ Levels of responses

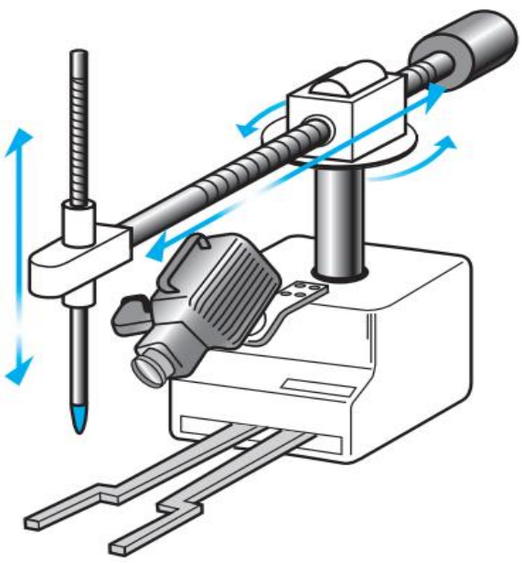
Levels of Responses

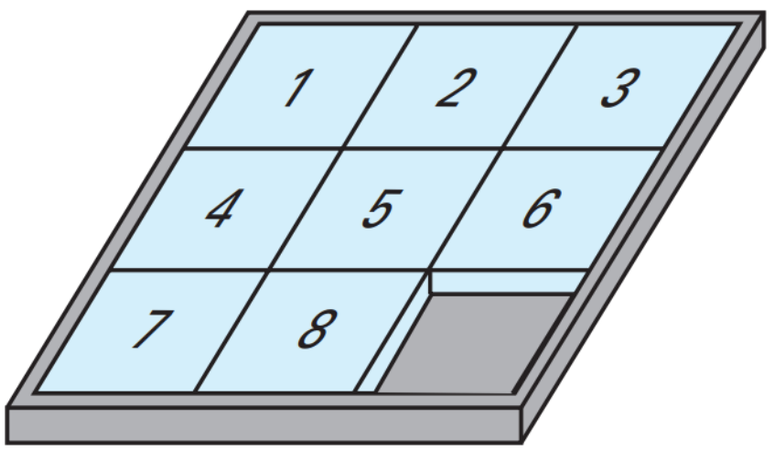
- ✓ Simplest response is a reflex action – predefined action
- ✓ Intelligent Actions – Adjust actions according to environment knowledge.
- ✓ The process of throwing a baseball is largely a reflex action but determining how and where to throw the ball requires knowledge of the current environment.

Levels of Responses

- ✓ In some cases, an agent responses improve over time as the agent learns. This could take the form of developing procedural knowledge (learning "how") or storing declarative knowledge (learning "what").

Example





Summary

Intelligent Agent

- ✓ Agent and Examples
- ✓ Data Acquisition
- ✓ Sensor
- ✓ Actuators
- ✓ Response Level
- ✓ Working Example

Topic-172. Artificial Intelligence: Research Methodologies

AI Domain

Pursued along two paths:

- ✓ Engineering Track
- ✓ Theoretical Track

Engineering Track

- ✓ Researchers are trying to develop systems that exhibit intelligent behavior

- ✓ Performance-oriented methodology
- ✓ To produce a product that meets certain performance goals

Theoretical Track

- ✓ Researchers are trying to develop a computational understanding of human –intelligence.
- ✓ Leads to a simulation-oriented methodology because the underlying goal is to expand our understanding of intelligence and thus the emphasis is on the underlying process rather than the exterior performance.

Example –Linguistic vs NLP

- ✓ Linguists are interested in learning how humans process language and thus tend toward more theoretical pursuits.
- ✓ Natural Language Processing (NLP) are interested in developing machines that can manipulate natural language and therefore lean in the engineering direction.

Example – building Shell for an OS

- ✓ Goal: Receive instructions from the outside world through verbal English commands.
- ✓ Shall need not to be aware of English language – e.g. finding all meaning of word “Copy” – verb or noun.
- ✓ Satisfactory for engineer.
- ✓ Not aesthetically pleasing to a theoretician.

Summary

AI: Research Methodologies

- ✓ Engineering Track
- ✓ Theoretical Track
- ✓ Examples

Topic-173. Artificial Intelligence: The Turing Test

The Turing Test

In the past the Turing test (proposed by Alan Turing in 1950) has served as a benchmark in measuring progress in the field of artificial intelligence.

How it worked

- ✓ Turing’s proposal was to allow a human, whom we call the interrogator, to communicate with a test subject by means of a typewriter system without being told whether the test subject was a human or a machine.
- ✓ The machine would be declared intelligent, if human remains unable to distinguish it from human.

Turing Prediction

- ✓ Turing predicted that by the year 2000 machines would have a 30 percent chance of passing a five-minute Turing test.
- ✓ That turned out to be surprisingly accurate.

Turing Test is not used today!

- ✓ This is due to the fact such intelligence has been found to be added with much easier:
- ✓ E.g. Doctor program
- ✓ “I am tired today,”
- ✓ When does not understand, “”Go on”, “Very interesting”

Turing Test is not used today!

- ✓ More recent examples of Turing test “successes” include Internet viruses that carry on “intelligent” dialogs with a human victim in order to trick the human into dropping his or her malware guard.
- ✓ **Turing Test is not used today!**
- ✓ Humans competing against the computer – e.g. chess, often experience the sensation that the machine possesses creativity and even a personality.

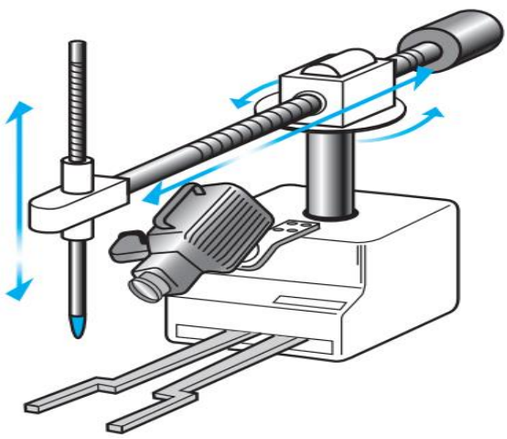
Summary

The Turing Test

- ✓ What is Turing Test
- ✓ Turing Predictions
- ✓ Why Turing Test is not used today?

Topic-174. Artificial Intelligence: Understanding Images

- ✓ The opening and closing of the gripper on the machine presents no serious obstacle.
- ✓ The ability to detect the presence of the puzzle in the gripper during this process is straight forward.
- ✓ Even the problem of focusing the camera on the puzzle can be handled simply by designing the gripper to position the puzzle at a particular predetermined position for viewing.
- ✓ Issue is understanding image.



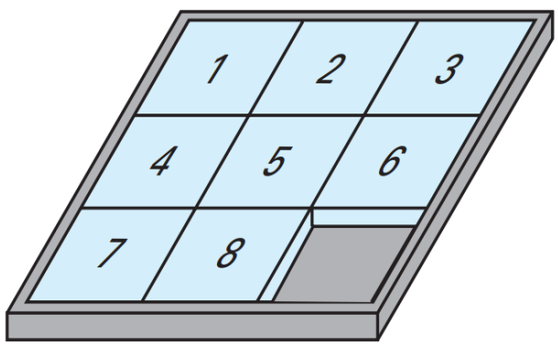


Image Processing – Simple Approach

- ✓ We can assume that what appears is always an image containing the digits 1 through 8 in a well-organized pattern.
- ✓ Picture of the puzzle has been encoded in terms of bits in the computer’s memory, with each bit representing the brightness level of a particular pixel.
- ✓ Our machine can detect which tile is in which position by comparing the different sections of the picture to prerecorded templates.

Matching Geometric Characteristics

- ✓ This technique of recognizing images is one method used in optical character readers.
- ✓ Drawback, requiring a certain degree of uniformity for the style, size, and orientation of the symbols being read.
- ✓ Will it work for handwritten material?
- ✓ The digit 1 might be characterized as a single vertical line.
- ✓ Digit 2 might be an opened curved line joined with a horizontal straight line across the bottom, and so on.

How it works?

Involves two steps:

- ✓ The first is to extract the features from the image being processed.
- ✓ The second is to compare the features to those of known symbols.

Process of understanding Images

Image Processing:
Refers to identifying characteristics of the
image known symbols.

Image Analysis: refers to the process of understanding what these characteristics mean.

Summary

Understanding Images

- ✓ Puzzle Example
- ✓ Pixel Based Approach
- ✓ Geometric Approach

Topic-175. Artificial Intelligence: Language Processing

Perception

- ✓ The success obtained in translating formal high-level programming languages into machine language led early researchers to believe that the ability to program computers to understand natural language was only a few years away.

Reality!

- ✓ Although formal language differ from natural languages like English, Urdu, German only a few years away.
- ✓ Programming languages are constructed from well-designed primitives so that each statement has only one grammatical structure and only one meaning.
- ✓ A statement in a natural language can have multiple meanings depending on its context or even the manner in which it is communicated.
- ✓ To understand natural language, humans rely heavily on additional knowledge and context.

Example

- ✓ I need an Apple!

Fruit or phone?

- ✓ “You are very late.”

Means speaker was waiting for a long time.

Understanding NLP

Requires:

- ✓ Syntactic Analysis
- ✓ Semantic Analysis
- ✓ Contextual Analysis

Syntactic Analysis

Major component is parsing.

“Mary gave John a birthday card.”
Subject = Mary

John got a birthday card from Mary.

Subject = John

Semantic Analysis

- ✓ Identifying the semantic role of each word in the statement.

- ✓ “Mary gave John a birthday card” and “John got a birthday card from Mary” would be recognized as saying the same thing.
- ✓ **Contextual Analysis**
- ✓ Context of the sentence is brought into the understanding process.
- ✓ The bat fell to the ground has different meanings in following two contexts:
    a) Baseball game
    b) Cave exploration

Summary

Language Processing

- ✓ Perception to Reality
- ✓ Understanding NLP
- ✓ Syntactic Analysis
- ✓ Semantic Analysis
- ✓ Contextual Analysis

Topic-176. CS Impact on Society

Impact Analysis

- ✓ CS has both positive and negative impacts on society.

Positive Impacts

- ✓ Improved communication
- ✓ Increased access to educational information via the internet
- ✓ Increased productivity
- ✓ Exponential business growth
- ✓ Faster response times (for example, email, instant messaging and chat)

Negative Impacts

- ✓ Ties to increased obesity
- ✓ Exposure to inappropriate material via the internet
- ✓ Increased crime and access to private information
- ✓ Hacking bank account and transfer money to your account
- ✓ Potential over-dependence on technology

Summary

CS Impact on Society

- ✓ Positive Impacts
- ✓ Negative Impacts

Topic-177. CS Impact on Health

Impact Analysis

- ✓ Computer Science has both positive and negative impacts on health.

Positive Impacts

- ✓ Easy access to online health centers
- ✓ Can take online appointment
- ✓ Can receive reports via email at home
- ✓ Hospitals can be managed more efficiently
- ✓ Online health tips/issues
- ✓ Health news

Negative Impacts

- ✓ Computer wrists – Typing too much
- ✓ Eyesight – Eye is not meant to look 2D for long amount of time
- ✓ Hearing loss
- ✓ Social isolation
- ✓ Communication issues
- ✓ Posture could lead bone issues
- ✓ Radiation damage – Airplane mode

Summary

CS Impact on Health

- ✓ Positive Impacts
- ✓ Negative Impacts

Topic-178. CS Impact on Environment

Impact Analysis

- ✓ Computer Science has both positive and negative impacts on environment.

Positive Impacts

- ✓ Paperless – Online reading and publishing.
- ✓ Americans use about 680 pounds of paper per person, per year.
- ✓ This amounts to about 85 million tons of paper or 2 billion trees.
- ✓ The average American household throws away 13,000 pieces of paper (around 1 billion trees in total in USA) each year. Most in the form of packaging and junk mail.

Negative Impacts

- ✓ In offices, people print unnecessarily.
- ✓ Computers are made from heavy metals (Lead, Mercury, Beryllium, Cadmium, PCV) when thrown away cause water contamination and air pollution.
- ✓ E-waste is sent to developing countries where people extract materials from these electronics such as Gold, Silver, and Copper.
- ✓ They extract them by burning the substances and that process releases hazardous smoke into the air.
- ✓ The people present around the e-waste gain diseases such as skin cancer, brain damage, lung cancer, kidney disease.
- ✓ Usage of energy.

Summary

CS Impact on environment

- ✓ Positive Impacts
- ✓ Negative Impacts

Topic-179. CS Impact: Ethical Issues

Ethical

- ✓ Relating to moral principles or the branch of knowledge dealing with these.

Major Issues

- ✓ Privacy
- ✓ Digital Ownership
- ✓ Data Gathering

Privacy

- ✓ Capacity to store, organized, and exchange records.
- ✓ Personal data, emails, passwords.
- ✓ Do people know how their accounts are monitored?
- ✓ To what extent is such monitoring occurring.

Digital Ownership

- ✓ Digital information is flowing more freely.
- ✓ How can ownership be established?
- ✓ Legal notions such as copyright have struggled to keep up with the digital era.

Data Gathering

- ✓ The United States has passed legislation allowing the government to actively monitor private citizens in the name of national security.
- ✓ What information can be gathered and why?
- ✓ Do people know what information is being monitored?
- ✓ Do they have a right to know how their data is being used?

Summary

Ethical Issues

- ✓ Privacy
- ✓ Digital Ownership
- ✓ Data Gathering

Topic-180. CS Impact: Software Licenses and Information Privacy

Software Licenses

- ✓ A software license is a document that provides legally binding guidelines for the use and distribution of software.
- ✓ Software licenses typically provide end users with the right to one or more copies of the software without violating copyrights.

Software Licensing Terms and Conditions

- ✓ Fair use of the software
- ✓ The limitations of liability
- ✓ Warranties and disclaimers
- ✓ Protections if the software or its use violates the intellectual property rights of others.

Software License Types

- ✓ Proprietary
- ✓ Free
- ✓ Open source

Information Privacy

- ✓ Data privacy, also called information privacy deals with the ability an organization or individual has to determine what data in a computer system can be shared with third parties.

Children's Online Privacy Protection Act

- ✓ (COPPA) - Gives parents control over what information websites can collect from their kids.

Health Insurance Portability and

Accountability Act

- ✓ (HIPPA) - Ensures patient confidentiality for all healthcare-related data.

Electronic Communications Privacy Act

- ✓ ECPA extends government restrictions on wire taps to include transmissions of electronic data.

Video Privacy Protection Act

- ✓ Prevents wrongful disclosure of an individual's personally identifiable information stemming from their rental or purchase of audiovisual material.

Gramm-Leach-Bliley Act

- ✓ Mandates how financial institutions must deal with the private information of individuals.

Summary

Software Licenses and Information Privacy

- ✓ Software License, Terms and Conditions and License Types
- ✓ Information Privacy and Laws

Topic-181. CS Impact: Intellectual Property

- ✓ Intellectual property (IP) is ideas, information and knowledge.
- ✓ 'Intellectual' because it is creative output and 'property' because it is viewed as a tradable product.
- ✓ Intellectual property rights (IPR) are specific legal rights which protect the owners of IP.

Patent

- ✓ A patent is a legal monopoly lasting 20 years granted in exchange for describing an invention and paying fees to the patent office.
- ✓ Patent is destroyed if disclosed before filing the application (except in UK for a short time).
- ✓ Think patent before publish.

Copyright

- ✓ Copyright is also the usual way of protecting software, although some software may be patented if it is a functional part of an invention.

Database Rights

- ✓ Database rights apply to databases which are not protected by copyright (an EU right only).

Design Rights

- ✓ Design rights applies to aspects of the shape or configuration of an article. Unregistered design right (for example, which covers computer chips).

Trademark

- ✓ A trademark is a mark (logo) or other distinctive sign applied to or associated with products or services.

Confidential Information

- ✓ Confidential information is knowledge which only you possess and which you have only revealed under a non-disclosure/confidentiality agreement.

IPR	Covers	Need to apply?	Maximum duration
Patent	Inventions	Yes	20 years
Copyright	Literary, musical, artistic works, and software	No	70 years after death of author
Registered design	Image – look and feel	Yes	25 years
Registered trade mark	Name, logo	Yes	Unlimited
Confidential information	Unpublished secret information	No	Unlimited
Database right	Databases	No	15 years

Summary

Intellectual property

- ✓ Definition
- ✓ IPR and its types
- ✓ Comparative Analysis

Topic-182. CS Impact: Security

- ✓ Computer security, also known as cyber security or IT security, is the protection of information systems from theft or damage to the hardware, the software, and to the information on them.

Security Areas

- ✓ Confidentiality
- ✓ Integrity
- ✓ Availability
- ✓ Authentication

Confidentiality

- ✓ Only authorized users can access the data resources and information.
  Authentication

Integrity

- ✓ Only authorized users should be able to modify the data when needed.

Availability

- ✓ Data should be available to users when needed.

Authentication

- ✓ Are you really communicating with whom you think you are communicating with?

Why its important

- ✓ Prevention of data theft such as bank account numbers, credit card information, passwords, work related documents or sheets, etc.

- ✓ Data present in a computer can also be misused by unauthorized intrusions.

Why its important

- ✓ Malicious intents can also be a factor in computer security. Intruders often use your computers for attacking other computers or websites or networks for creating havoc.

Firewall

- ✓ A firewall is a security-conscious piece of hardware or soft-ware that sits between the Internet and your network with a single-minded task: preventing them from getting to us.

Summary

Security

- ✓ Confidentiality
- ✓ Integrity
- ✓ Availability
- ✓ Authentication
- ✓ Firewall

Topic-183. CS Impact: Privacy

What is Privacy

- ✓ Individual's right to own the data generated by his or her life and activities, and to restrict the outward flow of that data.

Based on:

- ✓ Privacy uses the theory of natural rights, and generally responds to new information and communication technologies. In North America, Samuel D. Warren and Louis D. Brandeiswrote that privacy is the "right to be let alone“.

From where data comes from:

- ✓ Social Sites (Facebook, twitter)
- ✓ Telephone companies (mobile networks)
- ✓ Smart city cameras on road
- ✓ Emails
- ✓ Personal software
- ✓ Passwords, logins etc.

How to handle on internet

- ✓ Emails can be encrypted
- ✓ Anonymizing proxies
- ✓ Anonymizing networks

Privacy Information

- ✓ Google:

https://policies.google.com/privacy

- ✓ MSN:

https://www.msn.com/en-us/autos/partner/privacypolicy

- ✓ WhatsApp:

https://www.whatsapp.com/legal/#privacy-policy

EPIC

- ✓ Electronic Privacy Information Center
- ✓ Independent non-profit research center in Washington, D.C. Its mission is to focus public attention on emerging privacy and related human rights issues.

- ✓ EPIC has pursued several successful consumer privacy complaints with the US Federal Trade Commission, concerning Snapchat (faulty privacy Technology), WhatsApp (privacy policy after acquisition by Facebook (changes in user privacy settings), Google (roll-out of GoogleBuzz), Microsoft (Hailstorm log-in), issues.

PI

- ✓ Privacy International
- ✓ Charity organization of London, UK

Priority Areas:

- ✓ Challenging Data Exploitation
- ✓ Building the Global Privacy Movement
- ✓ Contesting Surveillance

Summary

Privacy

- ✓ Definition
- ✓ Data Points
- ✓ How to Handle
- ✓ International Privacy Control Bodies

Topic-184. CS Impact: Social Issues of IT

Issues

- ✓ Communication Breakdown
- ✓ Identity Theft
- ✓ Cyber Bullying
- ✓ Gaming Addiction
- ✓ Privacy
- ✓ Health & Fitness
- ✓ Plagiarism
- ✓ Terrorism
- ✓ Crime
- ✓ Access to Forbidden Literature

Summary

Social Issues of IT

- ✓ Discussion on important Social Impact of IT

Topic-185. CS Impact: Content Filtering, Email-Spams and Laws

Content Filtering

- ✓ Using a program to prevent access to certain items, which may be harmful if opened or accessed. The most common items to filter are executable, emails or websites.

How it works

- ✓ Content filtering works by matching strings of characters. When the strings match, the content is not allowed through. Content filters are often part of internet firewalls. In such a usage content filtering is serving a security purpose.
- ✓ Content filtering is also used to implement company policies related to information system usage. For example, it's common to filter websites like Facebook, YouTube etc. which are considered unrelated to work.

Spam emails

- ✓ As email spam continues to become a major issue, governments around the world have put specific regulations in place to protect their citizens from spams.

Important Laws

- ✓ CAN-SPAM Act in US
- ✓ Canada’s Anti-Spam Legislation (CASL)

- ✓ Anti-spam law in Europe
- ✓ Spam Act of 2003 in Australia
- ✓ Africa, Asia and South America have looser spam-law requirements

CAN-SPAM Act

- ✓ Don't use false or misleading header information
- ✓ Don't use deceptive subject lines
- ✓ Identify the message as an ad
- ✓ Tell recipients where you're located
- ✓ Honor opt-out requests promptly
- ✓ Monitor what others are doing on your behalf

Major requirements

- ✓ Ask for permission before adding emails
- ✓ User could identify you as sender
- ✓ Do not give email address to others
- ✓ Be honest

Summary

Content Filtering, Email-spams and Laws

- ✓ Discussion on important Social Impact of IT

Topic-186. CS Impact: Children Protection and Electronic Theft

Risks

- ✓ Content Risks
- ✓ Contact Risks
- ✓ Online Marketing
- ✓ Overspending
- ✓ Electronic Theft
- ✓ Information Privacy
- ✓ Information Security

Content Risks

- ✓ Illegal Content
- ✓ Harmful Content
- ✓ Harmful Advice

Contact Risks

- ✓ Online Harassment
- ✓ Cyberbullying
- ✓ Illegal Interactions
- ✓ Problematic Content Sharing

Online Marketing

- ✓ Inappropriate or unsuitable products
- ✓ Illegal and age-restricted products

Electronic Theft

- ✓ Online Fraud
- ✓ Online Scam
- ✓ Identify Theft

Information Privacy

- ✓ Personal data collection from children

Information Security

- ✓ Malicious Code
- ✓ Commercial Spyware

How to protect

- ✓ Web Filtering
- ✓ Parental Control Software
- ✓ Enable Safety Features
- ✓ Guide Kids

Summary

Children Protection and Electronic Theft

- ✓ Risks for Kids
- ✓ How to overcome these risks?

Topic-187. Word Processing: MS Word

Topics

- ✓ File Button
- ✓ Ribbon
- ✓ Dialogue Expander
- ✓ Mini Tool Bar

Topic-188. MS Word: Quick Access Bar

Topics

- ✓ Available options
- ✓ How to add commands to Quick Access Bar

Topic-189. MS Word: Home Ribbon

Topics

- ✓ Clipboard
- ✓ Font
- ✓ Paragraph
- ✓ Styles
- ✓ Editing

Topic-190. MS Word: Clipboard Group

Topics

- ✓ Cut
- ✓ Copy
- ✓ Paste
- ✓ Format Painter

Topic-191. MS Word: Font Group

Topics

- ✓ Font Families and Font Names
- ✓ Bold, Italic, and Underline
- ✓ Strikethrough
- ✓ Subscript and Superscript
- ✓ Text Effects
- ✓ Text Highlight Color
- ✓ Font Color
- ✓ Increase/Decrease Font Size
- ✓ Change Case

Topic-192. MS Word: Paragraph Group (Part-1)

Topics

- ✓ Alignment (Left, Center, Right, Justify)
- ✓ Line and Paragraph Space
- ✓ Shading
- ✓ Borders

Topic-193. MS Word: Paragraph Group (Part-2)

Topics

- ✓ Bullet Library
- ✓ Numbering
- ✓ Indentation

Topic-194. MS Word: Style Group

Topics

- ✓ Using Styles
- ✓ Creating Styles

Topic-195. MS Word: Editing Group

Topics

- ✓ Find
- ✓ Goto
- ✓ Replace
- ✓ Selection

Topic-196. MS Word: Insert Functionalities

Topics

- ✓ Pages Group
- ✓ Media
- ✓ Header & Footer
- ✓ Tables Group
- ✓ Links
- ✓ Text
- ✓ Illustrations Group
- ✓ Comments
- ✓ Symbols

Topic-197. MS Word: Page Group

Topics

- ✓ Cover Page
- ✓ Blank Page
- ✓ Page Break

Topic-198. MS Word: Table Group

Topics

- ✓ Insert Table
- ✓ Excel Spreadsheet
- ✓ Draw Table
- ✓ Quick Tables
- ✓ Convert Text to Table

Topic-199. MS Word: Illustration Group

Topics

- ✓ Pictures
- ✓ Shapes
- ✓ Chart
- ✓ Online Pictures
- ✓ SmartArt
- ✓ Screenshot

Topic-200. MS Word: Media and Links Groups

Topics

- ✓ Online Video
- ✓ Bookmark
- ✓ Hyperlink
- ✓ Cross-Reference

Topic-201. MS Word: Comments and Header & footer

Topics

- ✓ Comments
- ✓ Footer
- ✓ Header
- ✓ Page numbers

Topic-202. MS word: text group part-1

Topics

- ✓ Text Box
- ✓ Quick Parts
- ✓ Word Art

Topic-203. MS Word: Text Group (Part-2) and Symbols Group

Topics

- ✓ Drop Cap
- ✓ Object
- ✓ Equations and Symbols

Topic-204. MS word: Design ribbon

Topics

- ✓ Document Formatting
- ✓ Themes
- ✓ Colors
- ✓ Fonts
- ✓ Paragraph Spacing
- ✓ Watermarks
- ✓ Page Color
- ✓ Page Borders

Topic-206. MS Word: Paragraph Group in Page Layout Ribbon

Topics

- ✓ Margin
- ✓ Orientation
- ✓ Size
- ✓ Columns
- ✓ Breaks
- ✓ Line number
- ✓ Hyphenation

Topic-207. MS Word: Arrange Group in Page Layout Ribbon

Topics

- ✓ Position
- ✓ Wrap Text
- ✓ Bring Forward
- ✓ Send Backward
- ✓ Selection Pane
- ✓ Align
- ✓ Group
- ✓ Rotate

Topic-208. MS word: references ribbon

Topics

- ✓ Table of Contents
- ✓ Footnotes

Topic-209. MS Word: Proofing Group in Review Ribbon

Topics

- ✓ Spelling and Grammar
- ✓ Define
- ✓ Thesaurus
- ✓ Word Count

Topic-210. MS Word: Language Group in Review Ribbon

Topics

- ✓ Translate
- ✓ Language

Topic-211. MS Word: Comments Group in Review Ribbon

Topics

- ✓ New Comment
- ✓ Delete
- ✓ Previous
- ✓ Next
- ✓ Show Comments

Topic-212. MS Word: Tracking and Changes Groups in Review Ribbon

Topics

- ✓ Track Changes
- ✓ Markups
- ✓ Reviewing Pane
- ✓ Accept
- ✓ Reject
- ✓ Previous
- ✓ Next

Topic-213. MS Word: Compare and Protect Groups in Review Ribbon

Topics

- ✓ Compare
- ✓ Restrict Editing
- ✓ Password on saving

Topic-214. MS word: view ribbon

Topics

- ✓ Views Group
- ✓ Show Group
- ✓ Zoom Group
- ✓ Windows Group

Topic-215. MS PowerPoint: Introduction

Topics

- ✓ Why Presentation?
- ✓ Document and Presentation Difference
- ✓ Layout of main PowerPoint Window
- ✓ Resemblances with MS Word

Topic-216. MS PowerPoint: Slides Group on Home Ribbon

Topics

- ✓ New Slide
- ✓ Layout
- ✓ Reset
- ✓ Sections

Topic-217. MS PowerPoint: Design Ribbon

Topics

- ✓ Design Themes
- ✓ Slide Size
- ✓ Format background

Topic-218. MS PowerPoint: Transition Ribbon

Topics

- ✓ Transitions
- ✓ Sound
- ✓ Duration

Topic-219. MS PowerPoint: Animation Ribbon

Topics

- ✓ Animation
- ✓ Effect Options
- ✓ Add Animation

Topic-220. MS Excel: Introduction

Topics

- ✓ Introduction
- ✓ Cell and its address
- ✓ Format Cell
- ✓ Charts

Topic-221. MS excel: functions

Topics

- ✓ Sum
- ✓ Average
- ✓ Min
- ✓ Max
- ✓ Median
- ✓ Count
- ✓ CountIF
- ✓ Add
- ✓ Subtract

- ✓ Multiply
- ✓ Divide
- ✓ Exponents

Topic-222. MS Excel: Application Scenarios – I

Topics

- ✓ Practicing functions using Student Result Card application

Topic-223. MS Excel: Application Scenarios – II

Topics

- ✓ Practicing functions using Personal-expense management

Topic-224. MS Excel: Sorting and Filter

Topics

- ✓ Sorting
- ✓ Filter

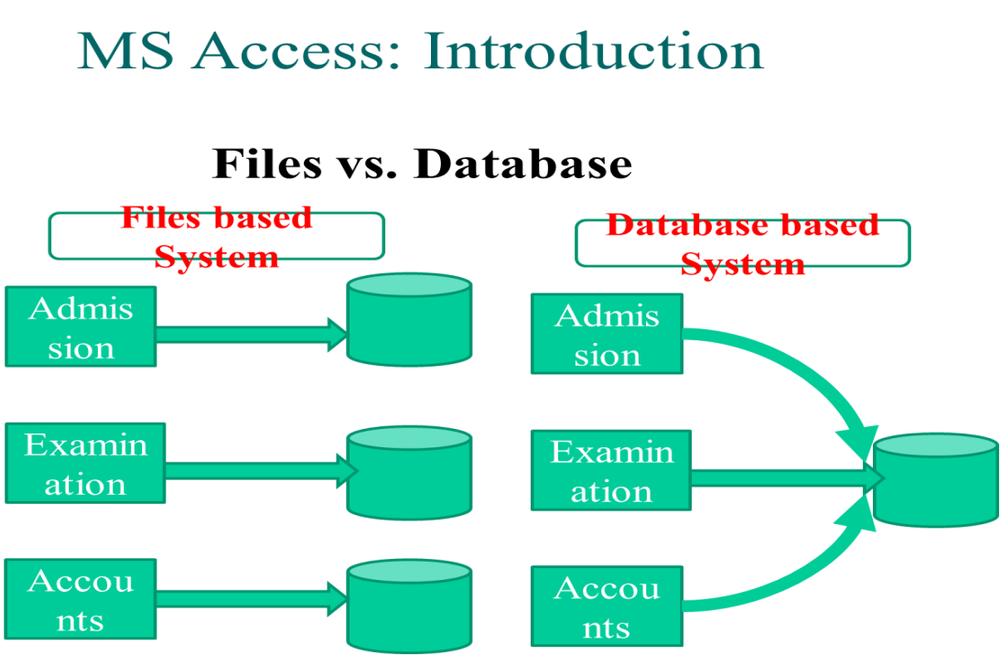
Topic-225. MS Access: Introduction

Topics

- ✓ Why Database?
- ✓ Ribbon Resemblances
- ✓ Creating a Database
- ✓ Importing Data from MS Excel

Definition

- ✓ A database is a shared collection of logically related data that is stored to meet the requirements of different users of an organization



MS Access: Introduction

Limitations of Files

Admission	Examination	Accounts
Registration Number	Registration Number	Registration Number
Name	Name	Name
Father Name	Father Name	Father Name
Admission Date	Exam Date	Fee status
Course of Study	Paper	Pending Installment

- ✓ Duplication of Data
- ✓ Data Dependence
- ✓ Limited number of built-in queries
- ✓ Update Anomalies -> Inconsistencies

MS Access: Introduction

Lets Start the practical

Summary

- ✓ What is MS Access
- ✓ Similarities of Ribbons
- ✓ Database creation and data import

Topic-226. MS Access: Creating and Managing Tables

Topics

- ✓ Create Table
- ✓ Add Fields
- ✓ Add Data
- ✓ Update and Delete

Summary Bar

- ✓ Experimented with Creating and managing tables

Topic-227. MS access: creating forms

Topics

- ✓ Create Forms
- ✓ Design Controls (Combo box, checkbox, Label, Image, Charts)
- ✓ Header and Footer (Data/Time, Logo, Title)
- ✓ Text Formatting in Home Ribbon

Summary Access Bar

- ✓ Experimented with Creating Forms

Topic-228. MS access: creating reports

Topics

- ✓ Create Reports
- ✓ Report Wizard
- ✓ Formatting

Summary Access Bar

- ✓ Experimented with report generation

Topic-229. MS Access: Query Wizard and Query Design

Topics

- ✓ Query Wizard Simple
- ✓ Query Wizard as summary
- ✓ Notepad Editor
- ✓ What is Web Page
- ✓ Using Notepad Editor to write web pages
- ✓ Code Complexities

Summary Access Bar

- ✓ Experimented with Query Wizard

Topic-230. Notepad editor

Topics

- ✓ Introduction to Dreamweaver
- ✓ Creating simple page, saving and viewing
- ✓ Properties Group

Summary Access Bar

- ✓ Experimented with Dreamweaver to write web page and properties group

Topic-231. Introduction to Dreamweaver

Topics

- ✓ Table Insertion and Code understanding

Summary Access Bar

- ✓ Experimented with Dreamweaver to Insert Tables

Topic-232. Inserting Tables using Dreamweaver

Topics

- ✓ Un-ordered Lists
- ✓ Ordered Lists

Summary Access Bar

- ✓ Experimented with Dreamweaver to Insert lists

Topic-233. Inserting lists

Topics

- ✓ Images Insertion
- ✓ Development of a web page

Summary Access Bar

- ✓ Experimented with Dreamweaver to insert images and development of web page

Topic-234. Inserting Images

Topics

- ✓ Images Insertion
- ✓ Development of a web page

Summary Access Bar

- ✓ Experimented with Dreamweaver to insert images and development of web page