

COMPUTER OPERATOR AND PROGRAMMING ASSISTANT

NSQF LEVEL - 3

TRADE THEORY

SECTOR: IT & ITES

(As per revised syllabus July 2022 - 1200 hrs)



Directorate General of Training

**DIRECTORATE GENERAL OF TRAINING
MINISTRY OF SKILL DEVELOPMENT & ENTREPRENEURSHIP
GOVERNMENT OF INDIA**



**NATIONAL INSTRUCTIONAL
MEDIA INSTITUTE, CHENNAI**

Post Box No. 3142, CTI Campus, Guindy, Chennai - 600 032

Sector : IT & ITES
Duration : 1 Year
Trade : COPA - Trade Theory - NSQF Level - 3 (Revised 2022)

Developed & Published by



National Instructional Media Institute
Post Box No.3142
Guindy, Chennai - 600 032
INDIA
Email: chennai-nimi@nic.in
Website: www.nimi.gov.in

Copyright © 2022 National Instructional Media Institute, Chennai

First Edition : January 2023 Copies : 2,000

Rs. 285/-

All rights reserved.

No part of this publication can be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopy, recording or any information storage and retrieval system, without permission in writing from the National Instructional Media Institute, Chennai.

FOREWORD

The Government of India has set an ambitious target of imparting skills to 30 crores people, one out of every four Indians, by 2020 to help them secure jobs as part of the National Skills Development Policy. Industrial Training Institutes (ITIs) play a vital role in this process especially in terms of providing skilled manpower. Keeping this in mind, and for providing the current industry relevant skill training to Trainees, ITI syllabus has been recently updated with the help of Mentor Councils comprising various stakeholder's viz. Industries, Entrepreneurs, Academicians and representatives from ITIs.

The National Instructional Media Institute (NIMI), Chennai has now come up with instructional material to suit the revised curriculum for **COPA Trade Theory NSQF Level - 3 (Revised 2022) in IT & ITES Sector**. The NSQF Level - 3 Trade Practical will help the trainees to get an international equivalency standard where their skill proficiency and competency will be duly recognized across the globe and this will also increase the scope of recognition of prior learning. NSQF Level - 3 trainees will also get the opportunities to promote life long learning and skill development. I have no doubt that with NSQF Level - 3 the trainers and trainees of ITIs, and all stakeholders will derive maximum benefits from these IMPs and that NIMI's effort will go a long way in improving the quality of Vocational training in the country.

The Executive Director & Staff of NIMI and members of Media Development Committee deserve appreciation for their contribution in bringing out this publication.

Jai Hind

Addl. Secretary/Director General (Training)
Ministry of Skill Development & Entrepreneurship,
Government of India.

New Delhi - 110 001

PREFACE

The National Instructional Media Institute (NIMI) was established in 1986 at Chennai by then Directorate General of Employment and Training (D.G.E & T), Ministry of Labour and Employment, (now under Ministry of Skill Development and Entrepreneurship) Government of India, with technical assistance from the Govt. of the Federal Republic of Germany. The prime objective of this institute is to develop and provide instructional materials for various trades as per the prescribed syllabi (NSQF LEVEL - 3) under the Craftsman and Apprenticeship Training Schemes.

The instructional materials are created keeping in mind, the main objective of Vocational Training under NCVT/NAC in India, which is to help an individual to master skills to do a job. The instructional materials are generated in the form of Instructional Media Packages (IMPs). An IMP consists of Theory book, Practical book, Test and Assignment book, Instructor Guide, Audio Visual Aid (Wall charts and Transparencies) and other support materials.

The trade practical book consists of series of exercises to be completed by the trainees in the workshop. These exercises are designed to ensure that all the skills in the prescribed syllabus are covered. The trade theory book provides related theoretical knowledge required to enable the trainee to do a job. The test and assignments will enable the instructor to give assignments for the evaluation of the performance of a trainee. The wall charts and transparencies are unique, as they not only help the instructor to effectively present a topic but also help him to assess the trainee's understanding. The instructor guide enables the instructor to plan his schedule of instruction, plan the raw material requirements, day to day lessons and demonstrations.

In order to perform the skills in a productive manner instructional videos are embedded in QR code of the exercise in this instructional material so as to integrate the skill learning with the procedural practical steps given in the exercise. The instructional videos will improve the quality of standard on practical training and will motivate the trainees to focus and perform the skill seamlessly.

IMPs also deals with the complex skills required to be developed for effective team work. Necessary care has also been taken to include important skill areas of allied trades as prescribed in the syllabus.

The availability of a complete Instructional Media Package in an institute helps both the trainer and management to impart effective training.

The IMPs are the outcome of collective efforts of the staff members of NIMI and the members of the Media Development Committees specially drawn from Public and Private sector industries, various training institutes under the Directorate General of Training (DGT), Government and Private ITIs.

NIMI would like to take this opportunity to convey sincere thanks to the Directors of Employment & Training of various State Governments, Training Departments of Industries both in the Public and Private sectors, Officers of DGT and DGT field institutes, proof readers, individual media developers and coordinators, but for whose active support NIMI would not have been able to bring out this materials.

Chennai - 600 032

EXECUTIVE DIRECTOR

ACKNOWLEDGEMENT

National Instructional Media Institute (NIMI) sincerely acknowledges with thanks for the co-operation and contribution extended by the following Media Developers and their sponsoring organisations to bring out this Instructional Material (**Trade Theory**) for the trade of **COPA** (NSQF LEVEL - 3) (Revised 2022) under **IT & ITES** Sector for ITIs.

MEDIA DEVELOPMENT COMMITTEE MEMBERS

Smt. M. Banumathy	-	Assistant Training Officer, Govt ITI, Ambattur.
Smt. V. Revathi	-	Assistant Training Officer, Govt ITI, Coimbatore.

NIMI CO-ORDINATORS

Shri.Nirmalya Nath	-	Deputy Director, NIMI- Chennai - 32.
Shri.G. Michael Johny	-	Manager, NIMI, Chennai - 32

NIMI records its appreciation for the Data Entry, CAD, DTP operators for their excellent and devoted services in the process of development of this Instructional Material.

NIMI also acknowledges with thanks the invaluable efforts rendered by all other NIMI staff who have contributed towards the development of this Instructional Material.

NIMI is also grateful to everyone who has directly or indirectly helped in developing this Instructional Material.

INTRODUCTION

TRADE PRACTICAL

The trade practical manual is intended to be used in workshop . It consists of a series of practical exercises to be completed by the trainees during the course of the **COPA** Trade supplemented and supported by instructions/informations to assist in performing the exercises. These exercises are designed to ensure that all the skills in compliance with NSQF LEVEL - 3

- | | |
|---|--|
| Module 1 - Safe Working Practices | Module 22 - Manage Tables and Bulleted Text |
| Module 2 - Assemble a Desktop PC | Module 23 - Manage Graphic Elements |
| Module 3 - Using Windows Operating System | Module 24 - Manage Audio & Video Elements |
| Module 4 - Computer Basics & Software Installation | Module 25 - Manage Transitions and Animations |
| Module 5 - DOS Command Line Interface | Module 26 - Manage Collaboration |
| Module 6 - Install Ubuntu Linux operating system and execute basic Linux commands | Module 27 - Demonstrate on MySQL |
| Module 7 - Using Word Processing Software | Module 28 - Demonstrate on Queries |
| Module 8 - Format documents | Module 29 - Demonstrate on Functions |
| Modfule 9 - Manage Tables and Lists | Module 30 - Set-up & Configure a Computer Network |
| Module 10 - Create and Manage References | Module 31 - Create Simple Static Web Pages using HTML Tags |
| Module 11 - Manage Graphic Elements | Module 32 - JavaScript Embed JavaScript in HTML Pages |
| Module 12 - Manage Document Collaboration | Module 33 - Data Visualization or Analysis using Excel |
| Module 13 - Manage Mailings | Module 34 - Browse E-Commerce Sites to Identify Products & Services |
| Module 14 - Spread Sheet Application, Manage Worksheets and Workbooks | Module 35 - Protect Information, Computers and Networks from Viruses, Spyware and other Malicious Code |
| Module 15 - Manage Data Cells and Ranges | Module 36 - Cloud Computing |
| Module 16 - Manage Tables and Table Data | Module 37 - Develop an application and perform the Application Development Life Cycle |
| Module 17 - Perform Operations using Formulas and Functions | Module 38 to 42 - Elective Module I - Programming in Python |
| Module 18 - Manage Charts | Module 38 to 42 - Elective Module II - Programming in JAVA |
| Module 19 - Manage Pivot Tables | |
| Module 20 - Power Point Presentations | |
| Module 21 - Format Presentations | |

The skill training in the shop floor is planned through a series of practical exercises centred around some practical project. However, there are few instances where the individual exercise does not form a part of project.

While developing the practical manual a sincere effort was made to prepare each exercise which will be easy to understand and carry out even by below average trainee. However the development team accept that there is a scope for further improvement. NIMI, looks forward to the suggestions from the experienced training faculty for improving the manual.

TRADE THEORY

The manual of trade theory consists of theoretical information for the course of the **COPA** Trade. The contents are sequenced according to the practical exercise contained in the manual on Trade practical. Attempt has been made to relate the theoretical aspects with the skill covered in each exercise to the extent possible. This co-relation is maintained to help the trainees to develop the perceptual capabilities for performing the skills.

The Trade theory has to be taught and learnt along with the corresponding exercise contained in the manual on trade practical. The indicating about the corresponding practical exercise are given in every sheet of this manual.

It will be preferable to teach/learn the trade theory connected to each exercise atleast one class before performing the related skills in the shop floor. The trade theory is to be treated as an integrated part of each exercise.

The material is not the purpose of self learning and should be considered as supplementary to class room instruction.

CONTENTS

Lesson No.	Title of the Lesson	Learning Outcome	Page No.
1.1.01&02 1.1.03&04	Module 1 : Safe Working Practices Electrical safety Safety practice - fire extinguishers	1	1 2
1.2.05&06	Module 2 : Assemble a Desktop PC Introduction to computers	1	7
1.3.07-13	Module 3 : Using Windows Operating System Introduction to CPU architecture and motherboard	1	16
1.4.14 1.4.15 1.4.16 1.4.17 1.4.18 1.4.19 1.4.20&21	Module 4 : Computer Basics & Software Installation View the BIOS settings and their modifications Install Windows operating system Format hard disk and create partition Identify and rectify common hardware and software issues during OS installation Install necessary application software for Windows i.e. Office Package, PDF Reader, Media Player etc Configure Bluetooth and Wi-Fi settings DVDs, CDs and burning DVDs	1 	21 29 31 35 39 40 42
1.5.22 1.5.23	Module 5 : DOS Command Line Interface Use basic DOS commands for directory listing Manage files and folders using DOS commands	1	44 47
1.6.24&25 1.6.26-29	Module 6 : Install Ubuntu Linux operating system and execute basic Linux commands Introduction to Linux operating system Handling commands and various editors	1	51 55
1.7.30-33	Module 7 : Using Word Processing Software MS WORD 2010 THEORY	2	79
1.8.34-36	Module 8 : Format documents Insert, format text and paragraphs, Create and configure document sections	2	88
1.9.37-39	Module 9 : Manage Tables and Lists Create, modify tables	2	90
1.10.40&41	Module 10 : Create and Manage References Create and manage reference elements and tables	2	92
1.11.42-45	Module 11 : Manage Graphic Elements Insert, format illustrations and text boxes	2	93
1.12.46&47	Module 12 : Manage Document Collaboration Manage comments change tracking and mailings	2	94

Lesson No.	Title of the Lesson	Learning Outcome	Page No.
1.13.48	Module 13 : Manage Mailings Perform mail merge	2	98
1.14.49-54	Module 14 : Spread Sheet Application, Manage Worksheets and Workbooks Open files in MS Excel	3	101
1.15.55-57	Module 15: Manage Data Cells and Ranges Manipulate data	3	102
1.16.58-60	Module 16: Manage Tables and Table Data Create and format tables	3	104
1.17.61-63	Module 17: Perform Operations using Formulas and Functions Functions and formulas in MS-Excel 2010	3	105
1.18.64-66	Module 18: Manage Mailings Manage Charts	3	110
1.19.67	Module 19: Manage Pivot Tables Create Pivot Tables	3	112
1.20.68-74	Module 20: Power Point Presentations Open files in MS PowerPoint Presentations	4	113
1.21.75-77	Module 21: Format Presentations Insert, Format text and paragraphs	4	114
1.22.78-80	Module 22: Manage Tables and Bulleted Text Create tables, modify tables, modify bulleted text	4	115
1.23.81-83	Module 23: Manage Graphic Elements Insert illustrations, Format illustrations and text boxes	4	116
1.24.84&85	Module 24: Manage Audio & Video Elements Audio & Video Elements	4	118
1.25.86&87	Module 25: Manage Transitions and Animations Add slide transitions & animations	4	119
1.26.88	Module 26: Manage Collaboration Add and manage comments	4	120
1.27.89-91 1.27.92-96	Module 27: Demonstrate on MySQL Install, Troubleshoot, Create and Use of database in MySQL Designing database using normalization rules, various datatypes, data integrity, DDL, DML & DCL Statements Enforcing Primary key and Foreign key	5	121 123
1.28.97&98	Module 28: Demonstrate on Queries Insert and delete queries Update queries	5	125
1.29.99&100	Module 29: Demonstrate on Functions Using the Number, Date and Character functions, group by having, sub query	5	126

Lesson No.	Title of the Lesson	Learning Outcome	Page No.
1.30.101-112	Module 30: Set-up & Configure a Computer Network Connect a computer to a network and share Devices i.e. Printers, files, folders and drives	6	128
1.31.113 1.31.114	Module 31: Create Simple Static Web Pages using HTML Tags Web designing Introduction to CMS and web authoring tools	7	137 156
1.32.115 1.32.116&117 1.32.118 1.32.119 1.32.120 1.32.121	Module 32: JavaScript Embed JavaScript in HTML Pages Understanding JavaScript Using JavaScript Variable and data types Control statements, Loops and Popup boxes in JavaScript Arrays in JavaScript Develop dynamic HTML pages using JavaScript Deploy web project using IIS	8	158 162 169 175 182 198
1.33.122-129	Module 33: Data Visualization or Analysis using Excel Create advanced formulas and macros	9	205
1.34.130-132 1.34.133 1.34.134-137	Module 34: Browse E-Commerce Sites to Identify Products & Services E-Commerce scope and benefits Undertake transactions on an e-commerce site E-Commerce Security issues and Payment Gateways	10	211 213 214
1.35.138 1.35.139	Module 35: Protect Information, Computers and Networks from Viruses, Spyware and other Malicious Code Overview of information security and threats Privacy Protection and IT Act	11	216 245
1.36.140-143	Module 36: Cloud Computing Working with Cloud Services	12	250
1.37.144&145	Module 37: Develop an application and perform the Application Development Life Cycle Identify Phases of the Application Development Life Cycle	12	254
1.38.01&02 1.39.03-05 1.40.06&07 1.41.08&09 1.42.10&11	Module 38 to 42 : Elective Module I - Programming in Python Programming language (Python) Use Python from command line Perform Operations using Data Types and Operators Control Flow with Decisions and Loops Document and Structure Code Perform Operations Using Modules and Tools	13	258 260 264 267 283
1.38.01-03 1.39.04-07 1.40.08-16 1.41.17-23 1.42.24-30	Module 38 to 42: Elective Module II Programming in JAVA Object Oriented Programming and JAVA Language Demonstrate writing JAVA programs JAVA Program Flow Control JAVA Classes, Overloading and Inheritance Abstract Classes and Interfaces in JAVA	14	287 291 304 309 316

LEARNING / ASSESSABLE OUTCOME

On completion of this book you shall be able to

S.No.	Learning Outcome	Ref. Ex.No.
1	Install and setup operating system and related software in a computer following safety precautions. (Mapped NOS: SSC/N3022)	1.1.01 - 1.6.29
2	Create, format, and edit document using word processing application software.(Mapped NOS: SSC/N3022)	1.7.30 - 1.13.48
3	Create, format, edit and develop a workbook by using spreadsheet application software. (Mapped NOS: SSC/N3022)	1.14.49 - 1.19.67
4	Create and customize slides for presentation. (Mapped NOS: SSC/N3022)	1.20.68 - 1.26.88
5	Create and manage database file by using MySQL. (NOS: SSC/N9401)	1.27.89 - 1.29.100
6	Install, setup/configure, troubleshoot, and secure computer network including Internet. (Mapped NOS: SSC/N3022)	1.30.101 - 1.30.112
7	Develop web pages using HTML and CSS. (Mapped NOS: SSC/N0503, SSC/N0501)	1.31.113 - 1.31.114
8	Develop web pages using JavaScript. (Mapped NOS: SSC/N0503, SSC/N0501)	1.32.115 - 1.32.121
9	Create workbooks with advanced formulas, macros, charts, pivot tables and demonstrate ability to use Power tools. (NOS: SSC/N9402)	1.33.122 - 1.33.129
10	Browse, select, and transact using E-commerce websites (NOS: SSC/N9403)	1.34.130 - 1.34.137
11	Secure information from Internet by using cyber security concept. (NOS: SSC/N9404)	1.35.138 - 1.35.139
12	Explain Cloud concepts &services and Describe Application Development Life Cycle. (NOS: SSC/N9405)	1.36.140 - 1.37.145
13	Write programs using Python language. (NOS: SSC/N9406)	1.38.01 - 1.42.11
14	Writing programs using JAVA. (SSC/N9407)	1.38.01 - 1.42.30

SYLLABUS

Duration	Reference Learning Outcome	Professional Skills (Trade Practical) with Indicative hours	Professional Knowledge (Trade Theory)
Professional Skill- 94 Hrs; Professional Knowledge - 32 Hrs	Install and setup operating system and related software in a computer following safety precautions. (Mapped NOS: SSC/N3022)	Safe working practices (10 Hrs) <ol style="list-style-type: none"> Visit COPA Lab. of the institutes and locate the electrical connections with computer system setup. (3 Hrs) Identifying safety symbols and hazard identification. (3 Hrs) Practice safe methods of fire fighting in case of electrical fire. (2 Hrs) Use of fire extinguishers. (2Hrs) Assemble a Desktop PC (8 hrs) <ol style="list-style-type: none"> Identify computer peripherals and internal components of a desktop computer. (4 Hrs) Assemble components of desktop computer. (4 Hrs) Using Windows Operating Systems (20 hrs) <ol style="list-style-type: none"> Practice on Windows interface and navigating windows. (3 Hrs) 	Introduction to Computers (3 Hrs) <ul style="list-style-type: none"> Safe working practices Scope of the COPA trade. Safety rules and safety signs. Types and working of fire extinguishers. Introduction to Computer components Introduction to computer system (4 Hrs) <ul style="list-style-type: none"> Concepts of Hardware and Software. Function of mother board components and various processors. Various Input/ Output devices in use and their features Introduction Windows Operating System (9 Hrs)

	<p>8. Practice on managing files and folders using removable drives. (4 Hrs)</p> <p>9. Customize the desktop (2 hrs)</p> <p>10. Settings and manage user accounts. (1 Hr)</p> <p>11. View system properties and control panel details. (3 Hrs)</p> <p>12. Work with keyboard shortcut commands. (4 Hrs)</p> <p>13. Print and scan document using different commands. (3 Hrs)</p> <p>Computer basics and Software Installation (20 Hrs)</p> <p>14. View the BIOS settings and their modifications. (3 Hrs)</p> <p>15. Install Windows operating system. (4 Hrs)</p> <p>16. Format hard disk and create partition. (3 Hrs)</p> <p>17. Identify and rectify common hardware and software issues during OS installation. (3 Hrs)</p> <p>18. Install necessary application software for Windows i.e. Office Package, PDF Reader, Media Player etc. (2 Hrs)</p> <p>19. Configure Bluetooth and Wi-Fi settings. (1 Hr)</p> <p>20. Install Drivers for printer, scanner, webcam and DVD etc. (2 Hrs)</p> <p>21. Burn data, video and audio files on CD/DVD using application software. (2 Hrs)</p> <p>DOS Command Line Interface (9Hrs)</p> <p>22. Use basic DOS commands for directory listing. (5 Hrs)</p> <p>23. Manage files and folders using DOS commands. (4 Hrs)</p> <p>Install Ubuntu Linux operating system and execute basic Linux commands (27 Hrs)</p> <p>24. Installation of Ubuntu Linux operating system (6 Hrs)</p> <p>25. Install necessary application software for Linux i.e. Office Package, PDF Reader, Media Player etc. (4 Hrs)</p> <p>26. Use Basic Linux commands for directory listing, file and folder management, password etc. (6 Hrs)</p> <p>27. Use the Linux graphical user interface for file and folder management, exploring the system etc. (6 Hrs)</p> <p>28. Customize desktop settings and manage user accounts in Linux. (3 Hrs)</p>	<ul style="list-style-type: none"> • Introduction to operating System • Main features of Windows OS • Concept of various shortcut commands. <p>Introduction to the booting process (6 Hrs)</p> <ul style="list-style-type: none"> • Introduction to various types of memories and their features. • Basic Hardware and software issues and their solutions. • Usage of Application software and Antivirus. <p>Introduction to DOS Command Line Interface & Linux Operating Systems (10 Hrs)</p> <ul style="list-style-type: none"> • Introduction to basic DOS Internal and External Commands. • Introduction to Open Source Software • Introduction to Linux Operating System features, structure, files and processes • Basic Linux commands.
--	--	---

		29. View system properties and manage system setting in Linux. (2 Hrs)	
Professional Skill - 47 Hrs.; Professional Knowledge - 14 Hrs	Create, format, and edit document using word processing application software. (Mapped NOS: SSC/N3022)	<p>Using Word Processing Software (47 hrs)</p> <p>Manage documents (11 Hrs.)</p> <p>30. Navigate within documents (2 Hrs)</p> <ul style="list-style-type: none"> • Search for text • Link to locations within documents • Move to specific locations and objects in documents • Show and hide formatting symbols and hidden text <p>31. Format documents (4.6 Hrs)</p> <ul style="list-style-type: none"> • Set up document pages • Apply style sets • Insert and modify headers and footers • Configure page background elements <p>32. Save and share documents (2 Hrs)</p> <ul style="list-style-type: none"> • Save documents in alternative file formats • Modify basic document properties • Modify print settings • Share documents electronically <p>33. Inspect documents for issues (2.4 Hrs)</p> <ul style="list-style-type: none"> • Locate and remove hidden properties and personal information • Locate and correct accessibility issues • Locate and correct compatibility issues <p>Format documents (8 Hrs.)</p> <p>34. Insert text and paragraphs (2 Hrs)</p> <ul style="list-style-type: none"> • Find and replace text • Insert symbols and special characters <p>35. Format text and paragraphs (3 Hrs)</p> <ul style="list-style-type: none"> • Apply text effects • Apply formatting by using Format Painter • Set line and paragraph spacing and indentation • Apply built-in styles to text • Clear formatting <p>36. Create and configure document sections (3 Hrs)</p> <ul style="list-style-type: none"> • Format text in multiple columns • Insert page, section, and column breaks • Change page setup options for a section <p>Manage tables and lists (9.5 Hrs)</p> <p>37. Create tables (3 Hrs)</p> <ul style="list-style-type: none"> • Convert text to tables • Convert tables to text 	<p>Using Word Processing Software (14 Hrs)</p> <ul style="list-style-type: none"> • Introduction to the various applications in MS office. • Introduction to Word features, Office button, toolbars. • Creating, saving and formatting and printing documents using Word. • Working with objects, macro, mail merge, templates and other tools in Word.

	<ul style="list-style-type: none"> • Create tables by specifying rows and columns <p>38. Modify tables (3 Hrs)</p> <ul style="list-style-type: none"> • Sort table data • Configure cell margins and spacing • Merge and split cells • Resize tables, rows, and columns • Split tables • Configure a repeating row header <p>39. Create and modify lists (3.5 Hrs)</p> <ul style="list-style-type: none"> • Format paragraphs as numbered and bulleted lists • Change bullet characters and number formats • Define custom bullet characters and number formats • Increase and decrease list levels • Restart and continue list numbering • Set starting number values Create and manage references (3 Hrs.) <p>40. Create and manage reference elements (1.4 Hrs)</p> <ul style="list-style-type: none"> • Insert footnotes and endnotes • Modify footnote and endnote properties • Create and modify bibliography citation sources • Insert citations for bibliographies <p>41. Create and manage reference tables (1.6 Hrs)</p> <ul style="list-style-type: none"> • Insert tables of contents • Customize tables of contents • Insert bibliographies <p>Manage graphic elements (8.5 Hrs.)</p> <p>42. Insert illustrations and text boxes (3 Hrs)</p> <ul style="list-style-type: none"> • Insert shapes • Insert pictures • Insert 3D models • Insert SmartArt graphics • Insert screenshots and screen clippings • Insert text boxes <p>43. Format illustrations and text boxes (3 Hrs)</p> <ul style="list-style-type: none"> • Apply artistic effects • Apply picture effects and picture styles • Remove picture backgrounds • Format graphic elements • Format SmartArt graphics • Format 3D models <p>44. Add text to graphic elements (1 Hr)</p> <ul style="list-style-type: none"> • Add and modify text in text boxes • Add and modify text in shapes " Add and modify SmartArt graphic content
--	--

	<p>45. Modify graphic elements (1.5 Hrs)</p> <ul style="list-style-type: none"> • Position objects • Wrap text around objects • Add alternative text to objects for accessibility <p>Manage document collaboration (3.5 Hrs.)</p> <p>46. Add and manage comments (1 Hrs)</p> <ul style="list-style-type: none"> • Add comments • Review and reply to comments • Resolve comments • Delete comments <p>47. Manage change tracking (2.5 Hrs)</p> <ul style="list-style-type: none"> • Track changes • Review tracked changes • Accept and reject tracked changes • Lock and unlock change tracking <p>Manage Mailings (3.5 Hrs)</p> <p>48. Perform mail merge (3.5 Hrs)</p> <ul style="list-style-type: none"> • Create envelopes • Create labels • Create a new mailing list • Perform mail merge using an existing list 	
Professional Skill - 72 Hrs.; Professional Knowledge - 18 Hrs	<p>Create, format, edit and develop a workbook by using spread sheet application software. (Mapped NOS: SSC/N3022)</p> <p>Spread Sheet Application (72 Hrs)</p> <p>Manage Worksheets and Workbooks (12 Hrs.)</p> <p>49. Open files in MS Excel (1.5 Hrs)</p> <ul style="list-style-type: none"> • Open MS Excel • Create a new Excel file • Create a new Excel file from a template • Open an existing Excel file <p>50. Import data (1.5 Hrs)</p> <ul style="list-style-type: none"> • Import data from txt files • Import data from csv files <p>51. Navigate within workbooks (2 Hrs)</p> <ul style="list-style-type: none"> • Search data • Navigate to named cells, ranges or workbook elements • Insert and remove hyperlinks <p>52. Format worksheets and workbooks (2 Hrs)</p> <ul style="list-style-type: none"> • Modify page setup • Adjust row height and column width • Customize headers and footers <p>53. Customize options and views (3 Hrs)</p> <ul style="list-style-type: none"> • Display and modify content in different views • Freeze rows and columns • Change window views • Modify basic workbook properties • Display formulas 	<p>Spread Sheet Application (18 Hrs)</p> <ul style="list-style-type: none"> • Introduction to Excel features and Data Types. • Cell referencing and linking Sheets. • Introduction to various functions in all categories of Excel. • Concepts of sorting, filtering and validating data. • Analyzing data using charts, data tables, pivot tables, goal seek and scenarios

	<p>54. Configure content for collaboration (2 Hrs)</p> <ul style="list-style-type: none"> • Set a print area • Save workbooks in alternative file formats • Configure print settings Manage data cells and ranges (12 Hrs.) <p>55. Manipulate data (3 Hrs)</p> <ul style="list-style-type: none"> • Paste data by using special paste options • Fill cells by using Auto Fill • Insert and delete multiple columns or rows • Insert and delete cells <p>56. Format cells and ranges (5 Hrs)</p> <ul style="list-style-type: none"> • Merge and Unmerge cells • Modify cell alignment, orientation and indentation • Format cells using Format Painter • Wrap text within cells • Apply number formats • Apply cell formats from the Format cells dialog box • Apply cell styles • Clear cell formatting <p>57. Define and reference named ranges (4 Hrs)</p> <ul style="list-style-type: none"> • Define a named range • Name a table • Summarize data visually Insert spark lines • Apply built-in conditional formatting • Remove conditional formatting <p>Manage tables and table data (12 Hrs.)</p> <p>58. Create and format tables (4 hrs)</p> <ul style="list-style-type: none"> • Create excel tables from cell ranges • Apply table styles • Convert tables to cell ranges <p>59. Manage tables and table data (5 Hrs)</p> <ul style="list-style-type: none"> • Add or remove table rows and columns • Configure table style options • Insert and configure total rows <p>60. Filter and sort table data (3 Hrs)</p> <ul style="list-style-type: none"> • Filter records • Sort data by multiple columns Perform operations using formulas and functions (12 Hrs.) <p>61. Insert references (3 Hrs)</p> <ul style="list-style-type: none"> • Insert relative, absolute and mixed references • Reference named ranges and named tables in formulas <p>62. Calculate and transform data (5 Hrs)</p> <ul style="list-style-type: none"> • Perform calculations using AVERAGE(), MIN(), MAX() and SUM() • Count cells by using COUNT(), COUNTIF() and COUNTBLANK() • Perform conditional operations by
--	--

	<p>using the IF() function</p> <p>63. Format and modify text(4 Hrs)</p> <ul style="list-style-type: none"> Format text using RIGHT(), LEFT() and MID() functions Format text using UPPER(), LOWER() and LEN() functions Format text using CONCAT() and TEXTJOIN() functions <p>Manage Charts (12 Hrs.)</p> <p>64. Create Charts (3 Hrs)</p> <ul style="list-style-type: none"> Create charts Create chart sheets <p>65. Modify charts (4 Hrs)</p> <ul style="list-style-type: none"> Add data series to charts Switch between rows and columns in source data Add and modify chart elements Add trend lines to chart <p>66. Format charts (5 Hrs)</p> <ul style="list-style-type: none"> Apply chart layouts Apply chart styles Add alternative text to charts for accessibility <p>Manage Pivot Tables (12 Hrs.)</p> <p>67. Create Pivot Tables(12 Hrs)</p> <ul style="list-style-type: none"> Create Pivot tables from cell ranges Manipulate fields (columns) to get desired analysis Use Filters for pivot tables Represent data as Count, Sum, Average & % of row / column Group data in Columns & rows for aggregate reports 	
Professional Skill - 53 Hrs; Professional Knowledge - 13 Hrs	<p>Create and customize slides for presentation. (Mapped NOS: SSC/N3022)</p> <p>Power point Presentations (9.5 Hrs.)</p> <p>68. Open files in MS PowerPoint (1 Hr)</p> <ul style="list-style-type: none"> Open MS PowerPoint Create a new PowerPoint file Create a new PowerPoint file from a template Open an existing PowerPoint file <p>69. Format Power Point Presentations (1.5 Hrs)</p> <ul style="list-style-type: none"> Add slides Add titles and text Select slide layouts Add PowerPoint templates Duplicate slides <p>70. Modify slide masters, handout masters, and note masters (2 Hrs)</p> <ul style="list-style-type: none"> Change the slide master theme or background Modify slide master content Modify slide layouts <p>71. Change presentation options and views (1.5 Hrs)</p> <ul style="list-style-type: none"> Change slide size Display presentations in different views 	<p>Power point Presentations (13 Hrs.)</p> <ul style="list-style-type: none"> Image editing, Presentations Introduction to Open Office. Introduction to the properties and editing of images. Introduction to different formats of images and their uses. Introduction to Power Point and its advantages. Creating Slide Shows. <p>Fine tuning the presentation and good presentation technique</p>

	<p>72. Save and share PowerPoint Presentations (1.5 Hrs)</p> <ul style="list-style-type: none"> • Save presentations in alternative file formats • Configure different Print settings • Share presentations electronically <p>73. Configure and present slide shows (1 Hr)</p> <ul style="list-style-type: none"> • Hide unwanted slides while presenting • Configure slide show options • Present slide shows by using Presenter View <p>74. Prepare presentations for collaboration (1 Hrs)</p> <ul style="list-style-type: none"> • Protect presentations by using passwords • Export presentations to other formats <p>Format presentations (6.5 Hrs.)</p> <p>75. Insert text and paragraphs (2 Hrs)</p> <ul style="list-style-type: none"> • Find and replace text • Insert symbols and special characters <p>76. Format text and paragraphs (2 Hrs)</p> <ul style="list-style-type: none"> • Apply text effects • Apply formatting by using Format Painter • Set line and paragraph spacing and indentation • Apply built-in styles to text <p>77. Create and configure sections (2.5 Hrs)</p> <ul style="list-style-type: none"> • Format text in multiple columns • Text and image presentation styles • Clear formatting <p>Manage tables and bulleted text (8 Hrs)</p> <p>78. Create tables (3 Hrs)</p> <ul style="list-style-type: none"> • Insert tables in PowerPoint • Apply built-in table styles • Create tables by specifying rows and columns <p>79. Modify tables (3 Hrs)</p> <ul style="list-style-type: none"> • Insert and delete table rows and columns • Configure cell margins and spacing • Merge and split cells • Resize tables, rows, and columns <p>80. Create and modify bulleted text (2 Hrs)</p> <ul style="list-style-type: none"> • Format paragraphs as numbered and bulleted lists • Change bullet characters and number formats • Increase and decrease list indents
--	--

	<ul style="list-style-type: none"> • Set starting number values • Restart and continue list numbering on different slides <p>Create and manage reference elements (hyperlinks) (1.25 Hrs)</p> <ul style="list-style-type: none"> • Create hyperlinks within presentations • Create hyperlinks in presentations for files and other sources <p>Manage graphic elements (11.5 Hrs)</p> <p>81. Insert illustrations and text boxes (3.5 Hrs)</p> <ul style="list-style-type: none"> • Insert shapes • Insert pictures • Insert SmartArt graphics • Insert screenshots and screen clippings <p>82. Format illustrations and text boxes (4 Hrs)</p> <ul style="list-style-type: none"> • Apply artistic effects • Apply picture effects and picture styles • Remove picture backgrounds • Crop images • Format graphic elements • Format SmartArt graphics <p>83. Add and modify text in graphic elements (4 Hrs)</p> <ul style="list-style-type: none"> • Add and modify text in text boxes • Add and modify text in shapes • Add and modify SmartArt graphic text • Create, insert and modify charts <p>Manage Audio & Video elements (6.5 Hrs)</p> <p>84. Add Audio elements (2 Hrs)</p> <ul style="list-style-type: none"> • Import audio files in presentations • Configure audio playback options <p>85. Add Video elements (4.5 Hrs)</p> <ul style="list-style-type: none"> • Import video files in presentations • Resize video to fit slide • Configure video playback options <p>Manage transitions and animations (9 Hrs)</p> <p>86. Add slide transitions (5 Hrs)</p> <ul style="list-style-type: none"> • Add same slide transition for all slides • Set transition effect duration • Configure transition start and finish options • Customise select slide transitions <p>87. Add animations (4 Hrs)</p> <ul style="list-style-type: none"> • Animate text and graphic elements • Order shapes, images, and text boxes • Group shapes, images, and text boxes 	
--	--	--

		<ul style="list-style-type: none"> Configure animation effects Configure animation paths Reorder animations on a slide <p>Manage collaboration (0.75 Hrs)</p> <p>88. Add and manage comments (0.75 Hrs)</p> <ul style="list-style-type: none"> Add comments Review and reply to comments 	
Professional Skill - 53 Hrs.; Professional Knowledge - 13 Hrs	Create and manage database file by using MySQL. (NOS: SSC/N9401)	<p>Demonstrate on (15 Hrs.)</p> <p>89. Installation of MySQL. (1 Hr)</p> <p>90. Troubleshooting basic installation issues. (1 Hr)</p> <p>91. Creation and use of database. (3 Hr)</p> <p>92. Designing of tables. (3 Hr)</p> <p>93. Applying data integrity rules. (2 Hr)</p> <p>94. Using the DDL, DCL and DML statements. (2 Hrs)</p> <p>95. Enforcing constraints, primary key and foreign key. (2 Hrs)</p> <p>96. Adding indices to Tables. (1 Hr)</p> <p>Demonstrate on (15 Hrs)</p> <p>97. Simple select queries. (5 Hrs)</p> <p>98. Insert and delete queries Update queries. (10 Hrs)</p> <p>Demonstrate on (23 Hrs)</p> <p>99. Using the Number, Date and Character functions. Joins and Functions (11.5 Hrs)</p> <p>100. Joins, Group by, Having, Sub query. (11.5 Hrs)</p>	<p>Database Concepts (4 Hrs.)</p> <ul style="list-style-type: none"> Concept of DBMS, RDBMS. Data Models, Concept of DBA, Database Users. Database Schema. Designing Database using Normalization Rules. Various data types Data integrity, DDL DML and DCL statements. Enforcing Primary key and foreign key. Adding Indices. <p>Queries (4 Hrs)</p> <ul style="list-style-type: none"> Concepts of Transactions ACID Property of Transaction Constraints. <p>Joins and Functions (5 Hrs)</p> <ul style="list-style-type: none"> Joining of tables Sub Queries " Functions used in query like sum, average, max, min, count etc.
Professional Skill - 68 Hrs; Professional Knowledge - 16 Hrs	Install, setup/configure, troubleshoot, and secure computer network including Internet. (Mapped NOS: SSC/N3022)	<p>Computer Network (68 Hrs.) Set-up & configure a Computer Network (48 Hrs.)</p> <p>101. View Network connections. (2 Hrs)</p> <p>102. Connect a computer to a network and share Devices i.e. Printers, files, folders and drives. (4 Hrs)</p> <p>103. Work with various Network devices, connectors and cables. Create straight and cross cable and punch a UTP cable in the patch socket and test the connectivity. (4 Hrs)</p> <p>104. Practice IP Addressing and Subnet masking for IPV4/ IPV6 and pinging to test networks. (4 Hrs)</p> <p>105. Configure Hub and Switch. (4 Hrs)</p> <p>106. Set up and configure wired and wireless LAN in a Computer Lab within at least three computers. (6 Hrs)</p> <p>107. Use patch panel & I/O Box for wired LAN and installing & configuring Internet connection in a single PC and in a LAN. (6 Hrs)</p> <p>108. Set up a proxy server/ DHCP Server with firewall. (8 Hrs)</p> <p>109. Set up video conferencing using open-source software. (4 Hrs)</p>	<p>Communicating in a Connected World (12 Hrs.)</p> <ul style="list-style-type: none"> Local Networks, Communicating on a Local Network, Principles of Communications, How do Ethernet Networks Work?, How are Networks Built?, Routing Across Networks Explain how end-user devices and local networks interact with the global Internet. Communicating in a Connected World Explain the concept of network communication. Local Networks Explain the roles of devices in a network. What Does a Home Network Look Like? How Does Wi-Fi Work? Introduction to LAN Devices, Internetworking Devices, <p>Internet Concepts (4 Hrs)</p> <ul style="list-style-type: none"> Introduction to www, Concept of Internet, Web Browsers, internet servers and search engines. Concepts of Domain naming Systems and E mail communication. Introduction to video chatting tools and Social Networking concepts.

		<p>110. Use various tools (by open source /free) for network troubleshooting, maintenance and security for both Wired and Wireless (6 Hrs)</p> <p>111. Set up Internet access & communication (10 Hrs)</p> <ul style="list-style-type: none"> • Set-up internet connectivity • Set-up digital communication <p>112. Use the Internet (10 Hrs)</p> <ul style="list-style-type: none"> • Browse the Internet • Use e-mail • Use Social Media • Use the phone for online activities 	
<p>Professional Skill - 67 Hrs;</p> <p>Professional Knowledge - 17 Hrs</p>	<p>Develop web pages using HTML and CSS. (Mapped NOS: SSC/N0503, SSC/N0501)</p>	<p>Create simple static web pages using HTML tags (67 Hrs.)</p> <p>113. Practice HTML (46 Hrs)</p> <ul style="list-style-type: none"> • Practice with basic HTML elements (e.g. head, title, body), tag and attributes. • Design simple web page with text, paragraph and line break using HTML tags • Format text, change background colour and insert pictures in web page • Design simple web page with tables and lists. • Use marquees, hyperlinks and mail to link in designing web pages • Create frames, add style and design layout. • Display a web page within a web page using iframes. • Insert text, check and combo box in web page. • Design web page using password field, submit button • Reset button and radio button etc. • Design a web page adding flash file, audio and video files. • Design web page with forms and form controls using HTML tags <p>114. Create simple static web pages using CSS (21 Hrs)</p> <ul style="list-style-type: none"> • CSS syntax, Adding colors, fonts, backgrounds, images borders, text alignment, text transformation, Lists etc. • 3 types of CSS • Adding a Navigation Bars(vertical/horizontal bars) • CSS drop downs & Forms • CSS counters and website layout, Multiple backgrounds & Putting the stylesheet in a separate file • CSS Animations & CSS Buttons 	<p>Web Design Concepts (17 Hrs)</p> <ul style="list-style-type: none"> • Concepts of Static and Dynamic Web pages. • Introduction to HTML and various tags in HTML. • Concepts of different controls used in Web Pages. • Concepts of CSS and applying CSS to HTML. • Introduction to open source CMS viz, Joomla, Word press etc. and Web authoring tools viz. Kompozer, WordPress, Front Page etc.
<p>Professional Skill - 173 Hrs;</p>	<p>Develop web pages using JavaScript. (Mapped NOS:</p>	<p>JavaScript (173 Hrs)</p> <p>Embed JavaScript in HTML Pages (127 Hrs)</p>	<p>Introduction to JavaScript (35 Hrs)</p> <ul style="list-style-type: none"> • Introduction to Programming and Scripting Languages.

Professional Knowledge - 35 Hrs	SSC/N0503, SSC/N0501)	<p>115. Practicing the JavaScript in creating dynamic HTML pages. (53 Hrs)</p> <p>116. Embed JavaScript in HTML to Display Information in Web pages. (31 Hrs)</p> <p>117. Use error handling techniques in JavaScript. (11 Hrs)</p> <p>118. Use objects and classes in JavaScript. (31 Hrs)</p> <p>119. Describe Animation and Multimedia using JavaScript. (1 Hr)</p> <p>Create a dynamic website using an open-source tool (40 Hrs)</p> <p>120. Develop dynamic HTML pages using JavaScript.</p> <p>Deploy a simple web project (6 Hrs)</p> <p>121. Deploy web project using IIS.</p>	<ul style="list-style-type: none"> • Introduction to JavaScript and its application for the web. • Introduction to Web Servers and their features. • JavaScript Basics - Data types, Variables, Constants and Conversion between data types. • Arithmetic, Comparison, Logical Operators in JavaScript. Operator precedence. • Program Control Statements and loops in JavaScript. • Arrays in JavaScript - concepts, types and usage. • The String data type in JavaScript. Introduction to String, Math and Date. • Introduction to Functions in JavaScript. • Built in JavaScript functions overview. • Concepts of Pop Up boxes in JavaScript. • Introduction to the Document Object Model. • Concepts of using Animation and multimedia files in Java Script.
Professional Skill - 73 Hrs Professional Knowledge - 17 Hrs	Create workbooks with advanced formulas, macros, charts, pivot tables and demonstrate ability to use Power tools. (NOS: SSC/N9402)	<p>Data Visualization or analysis using Excel - (73 Hrs)</p> <p>Create advanced formulas and macros (24 Hrs)</p> <p>122. Create and modify simple macros (6 Hrs)</p> <p>123. Perform form controls and create simple data entry form with macros. (6 Hrs)</p> <p>124. Look up data by using functions. (6 Hrs)</p> <p>125. Use advanced date functions.(6 Hrs)</p> <p>Manage advanced charts and tables (21 hrs)</p> <p>126. Create and modify advanced charts. (10 Hrs)</p> <p>127. Create and modify PivotTables. (11 Hrs)</p> <p>Use Power Query and Power BI (24 Hrs)</p> <p>128. Create a Power Query, Power Query Function. Invoking the Power Query function and combining queries. Organize the workbook queries (12 Hrs)</p> <p>129. Use Power BI for simple data visualizations. (12 Hrs)</p> <p>Make a dashboard in Excel (4 Hrs)</p>	<p>Advanced Excel Concepts - Theory-(17 Hrs)</p> <ul style="list-style-type: none"> • MS excel revision (row, columns, basic formatting, insert menu, Print setup, etc.) and Look up introduction and functions • Types of references and cell naming • Excel Linkage Custom Format and Excel Protection • Tips and tricks • Pivot table and Pivot chart • Conditional formatting • Advanced Graphs • Power Queries

Professional Skill - 25hrs; Professional Knowledge - 10 Hrs.	Browse, select, and transact using E-commerce websites (NOS: SSC/N9403)	Browse e-Commerce sites to identify products & services (6.5 Hrs) 130. Demonstrate e-Commerce sites. (1.5 Hrs) 131. List features of e-commerce sites. (2 Hrs) 132. Use e-commerce sites to source an item. (3 Hrs) Shop online (4.5 Hrs) 133. Undertake transactions on an e-commerce site. (4.5 Hrs) Manage e-commerce operations (14 Hrs) 134. Add products to an ecommerce website. (4 Hrs) 135. Practice order processing. (3 Hrs) 136. Practice payment processing. (4.5 Hrs) 137. Identify common security issues. (3.5 Hrs)	e-Commerce (10 Hrs) <ul style="list-style-type: none"> Introduction to E Commerce and advantages. Building business on the net. Payment and Order Processing, Authorization, Chargeback and otherpayment methods. Security issues and payment gateways.
Professional Skill - 20 Hrs Professional Knowledge - 10 Hrs.	Secure information from Internet by using cybersecurity concept. (NOS: SSC/N9404)	138. Protect information, computers and networks from viruses, spyware and other malicious code (19 Hrs) <ul style="list-style-type: none"> Explain Cyber security (2 Hrs) Secure computers & the network (5.5 Hrs) Reduce cyber security threats (2 Hrs) Secure a Wi-Fi Network (4 Hrs) Use Anti-Virus software (3 Hrs) Perform back-ups of files, data & information (2.5 Hrs) 139. Explain compliance with IT Act (1 Hr) <ul style="list-style-type: none"> Identify steps for information privacy. (0.5 Hrs) Identify common cybercrimes and penalties applicable. (0.5 Hrs) 	Cyber Security (10 Hrs) <ul style="list-style-type: none"> Overview of Information Security, SSL, HTTPS, Security threats, information Security vulnerability and Risk management. Introduction to Directory Services, Access Control, Security, Privacy protection, Audit and Security. Introduction to IT Act and penalties for cybercrimes.
Professional Skill -25 Hrs; Professional Knowledge 15 Hrs.	Explain Cloud concepts &services and Describe Application Development Life Cycle. (NOS: SSC/ N9405)	Cloud Computing (15 Hrs) Working with Cloud Services (12 Hrs) 140. Practice with IaaS using free cloud services. (4 Hrs) 141. Practice with PaaS using free cloud services. (4 Hrs) 142. Practice with SaaS using free cloud services. (4 Hrs) Web hosting in Cloud (3 Hrs) 143. Host a website in a free cloud. (3 Hrs) Develop an application and perform the Application Development Life Cycle (10 Hrs) 144. Identify Phases of the Application Development Life Cycle. (5 Hrs) 145. Describe Roles in each of the phases of Application Development Life Cycle. (5 Hrs)	Introduction to Cloud Computing (12 Hrs) <ul style="list-style-type: none"> Benefits of cloud services, different categories. Resources available in cloud. Explain the Application Development Life Cycle (3 Hrs) <ul style="list-style-type: none"> Identify Phases of the Application Development Life Cycle. Describe Roles in each of phases of the Application Development Life Cycle.

To be selected either "Programming in Python" or "Programming in Java" as Elective Module for COPA.

COPA - Elective Module - I Programming in Python			
Professional Skill - 70 Hrs; Professional Knowledge - 30 Hrs.	Write programs using Python language. (NOS: SSC/N9406)	<p>Programming language (Python) Use Python from command line (7 Hrs)</p> <ol style="list-style-type: none"> 1. Install, set up the environment & run Python. (3 Hrs) 2. Use Command Line and IDE to create and execute a python program. (4 Hrs) <p>Perform Operations using Data Types and Operators (15 Hrs)</p> <ol style="list-style-type: none"> 3. Write and test a python program to demonstrate print statement, comments, different types of variables. (5 Hrs) 4. Write and test a python program to perform data and data type operations, string operations, date, input and output, output formatting and operators. (5 Hrs) 5. Determine the sequence of execution based on operator precedence. (5 Hrs) <p>Control Flow with Decisions and Loops (20hrs)</p> <ol style="list-style-type: none"> 6. Construct and analyze code segments that use branching statements. (10 Hrs) 7. Construct and analyze code segments that perform iteration. (10 Hrs) <p>Document and Structure Code (18 Hrs)</p> <ol style="list-style-type: none"> 8. Document code segments using comments and documentation strings. (3 Hrs) 9. Construct and analyze code segments that include List comprehensions, tuple, set and Dictionary comprehensions. (16 Hrs) <p>Perform Operations Using Modules and Tools (10 Hrs)</p> <ol style="list-style-type: none"> 10. Perform basic operations using built-in modules. (5 Hrs) 11. Solve complex computing problems by using built-in modules. (5 Hrs) 	<p>Programming language (Python) (30 Hrs)</p> <ul style="list-style-type: none"> • Introduction to Python History • Features, Setting up path Basic Syntax, Comments, Variable • Different Data Types • Casting, string, Boolean • Python Operators • Conditional Statements • Looping • Control Statements, String Manipulation, Lists, Tuple, sets • Dictionaries • Arrays • Iterators, modules, dates, math, • Modules, Input and Output.

COPA - Elective Module - II Programming in JAVA

Professional Skill - 70 Hrs; Professional Knowledge - 30 Hrs.	Writing programs using JAVA. (SSC/N9407)	<p>Object Oriented Programming and JAVA Language (15 Hrs)</p> <ol style="list-style-type: none"> 1. Installing JAVA. 2. Setting the Class path. 3. Writing and Executing a simple JAVA Program to display "Hello". <p>Demonstrate writing JAVA programs:</p> <ol style="list-style-type: none"> 4. Use of various data types in JAVA. 5. Use of various operators in JAVA. 6. Create and use of Local, Instance and Class variables. 	<p>Explain the following:</p> <ul style="list-style-type: none"> • Object Oriented Programming with Core Java • Java Programming features • JVM, Byte codes and Class path • Java Program Development • Compilation and Execution of JAVA programs • Basic JAVA language elements - keywords, comments, data types and variables.
--	--	--	--

	<p>7. Read text from the keyboard using scanner class read text from the keyboard using console class.</p>	<ul style="list-style-type: none"> • JAVA Arithmetic, Assignment, Relational, Logical, Increment / Decrement operators and expressions. • JAVA String Operators • JAVA Input and Output streams, System in, System out. • Input using Scanner class and Console class methods. (10Hrs.)
	<p>JAVA Program Flow Control (20 Hrs) Demonstrate writing JAVA programs:</p> <p>8. Use of the if and if ... else statements. 9. Use of the Switch statement. 10. Use of the Do ... While and while - do loops. 11. Use of the For Loop. 12. Use of the Break and Continue Keywords. 13. Use of the JAVA Numbers Class methods. 14. Use of the JAVA Character Class methods. 15. Use of the JAVA String Class methods. 16. Create and use of arrays.</p>	<p>Explain the following:</p> <ul style="list-style-type: none"> • Decision making and flow control using if...then, if then else, nested if, switch case and the conditional ternary operators in JAVA. • Loop control flow using while - do, do - while loops, for loop, using the break, continue statements. • Terminating the JAVA program. <p>JAVA Number, Character and String Classes. Arrays in JAVA. (6Hrs.)</p>
	<p>JAVA Classes, Overloading and Inheritance (20 Hrs) Demonstrate writing JAVA programs:</p> <p>17. Create and use of simple classes, objects and methods in JAVA. 18. Pass data and Objects to Methods. 19. Return data and Objects from Methods. 20. use of constructors in JAVA. 21. Create and use of Overloaded methods in JAVA. 22. Override methods in JAVA. 23. Create and use of Super class, Sub class in JAVA.</p>	<p>Explain the following:</p> <ul style="list-style-type: none"> • JAVA Objects, Classes and Methods. • Passing data and objects as parameters to methods. • Method Overloading. • Constructors and Overloaded constructors. • Inheritance in JAVA. • Method Overriding in JAVA. (8Hrs.)
	<p>Abstract Classes and Interfaces in JAVA (15 Hrs) Demonstrate writing JAVA programs:</p> <p>24. Create and use virtual methods. 25. Create abstract classes and methods. 26. Create interfaces in JAVA. 27. Override methods in JAVA. 28. Create and implement an interface. 29. Extend interfaces in JAVA. 30. Create and use a package in JAVA.</p> <p>Troubleshooting Java issues Download and Install Java, Check and Verify Java Configurations, Test Java, Remove Old Versions of Java, Find Java version, Always redirected to the java.com download page</p>	<p>Explain the following:</p> <ul style="list-style-type: none"> • Concept of Virtual methods. • Concept of Abstract classes and methods • Features of Abstract Classes • JAVA Interfaces and their advantages • Method Overriding in JAVA • Polymorphism in JAVA • Creating, implementing and extending interfaces • Creating and using Packages in JAVA. (6Hrs.)

Industrial Visit/Project work / Broad Area:

- Create and host a web site of at least 6 web pages using JavaScript & CSS containing interactive objects, functions etc.
- Create a project with Excel on Payroll Systems.
- Create a database with MySQL on Library management system.
- Create project in Python/Java programming language.

Electrical safety

Objective : At the end of this lesson you shall be able to

- explain how to rescue a person who is in contact with a live wire.

The severity of an electric shock will depend on the level of current which passes through the body and the length of time of contact. Do not delay, act at once. Make sure that the electric current has been disconnected.

If the casualty is still in contact with the supply - break the contact either by switching off the power, removing the plug or wrenching the cable free. If not, stand on some insulating material such as dry wood, rubber or plastic, or using whatever is at hand to insulate yourself and break the contact by pushing or pulling the person free. (Figs 1 & 2)

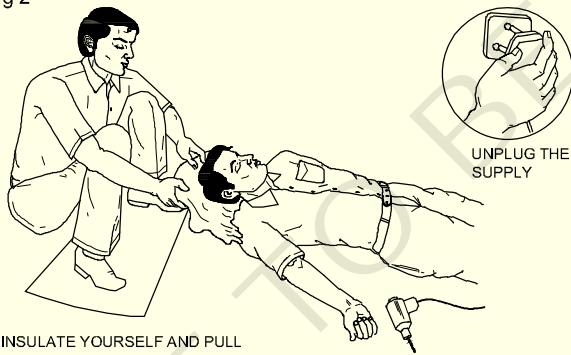
If you remain un-insulated, do not touch the victim with your bare hands until the circuit is made dead or person is moved away from the equipment.

Fig 1



EL110421

Fig 2



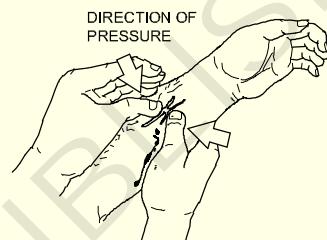
EL110422

If the victim is aloft, measures must be taken to prevent him from falling or atleast make him fall safe.

Electric burns on the victim may not cover a big area but may be deep seated. All you can do is to cover the area with a clean, sterile dressing and treat for shock. Get expert help as quickly as possible.

If the casualty is unconscious but is breathing, loosen the clothing about the neck, chest and waist and place the casualty in the recovery position. (Fig 3)

Fig 3



EL110433

Keep a constant check on the breathing and pulse rate.

Keep the casualty warm and comfortable. (Fig 4)

Send for help.

Do not give an unconscious person anything by mouth.

Do not leave an unconscious person unattended.

If the casualty is not breathing - Act at once - don't waste

Fig 4



EL110424

Safety practice - fire extinguishers

Objectives : At the end of this lesson you shall be able to

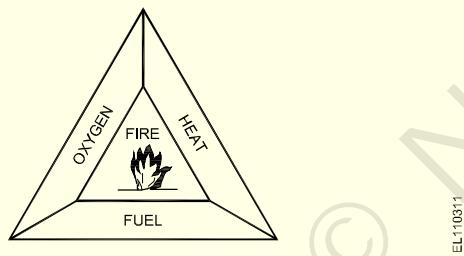
- explain how to rescue a person who is in contact with a live wire
- distinguish the different types of fire extinguishers
- determine the correct type of fire extinguisher to be used based on the class of fire
- describe the general procedure to be adopted in the event of a fire.

Fire is the burning of combustible material. A fire in an unwanted place and on an unwanted occasion and in an uncontrollable quantity can cause damage or destroy property and materials. It might injure people, and sometimes cause loss of life as well. Hence, every effort must be made to prevent fire. When a fire outbreak is discovered, it must be controlled and extinguished by immediate corrective action.

Is it possible to prevent fire? Yes, fire can be prevented by eliminating anyone of the three factors that causes fire.

The following are the three factors that must be present in combination for a fire to continue to burn. (Fig 1)

Fig 1



Fuel: Any substance, liquid, solid or gas will burn, if there is oxygen and high enough temperatures.

Heat: Every fuel will begin to burn at a certain temperature. It varies and depends on the fuel. Solids and liquids give off vapour when heated, and it is this vapour which ignites. Some liquids do not have to be heated as they give off vapour at normal room temperature say 15°C, eg. petrol.

Oxygen: Usually exists in sufficient quantity in air to keep a fire burning.

Extinguishing of fire: Isolating or removing any of these factors from the combination will extinguish the fire. There are three basic ways of achieving this.

- **Starving** the fire of fuel removes this element.
- **Smothering** - ie. isolate the fire from the supply of oxygen by blanketing it with foam, sand etc.
- **Cooling** - use water to lower the temperature. Removing any one of these factors will extinguish the fire.

Preventing fires: The majority of fires begin with small outbreaks which burn unnoticed until they have a secure hold. Most fires could be prevented with more care and by following some simple common sense rules.

Accumulation of combustible refuse (cotton waste soaked with oil, scrap wood, paper, etc.) in odd corners are a fire risk. Refuse should be removed to collection points.

The cause of fire in electrical equipment is misuse or neglect. Loose connections, wrongly rated fuses, overloaded circuits cause overheating which may in turn lead to a fire. Damage to insulation between conductors in cables causes fire.

Clothing and anything else which might catch fire should be kept well away from heaters. Make sure that the heater is shut off at the end of the working day.

Highly flammable liquids and petroleum mixtures (thinner, adhesive solutions, solvents, kerosene, spirit, LPG gas etc.) should be stored in the flammable material storage area.

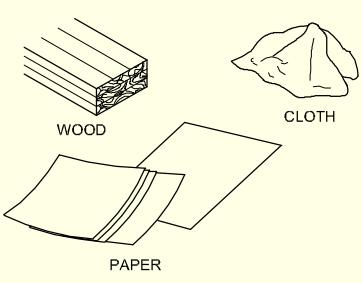
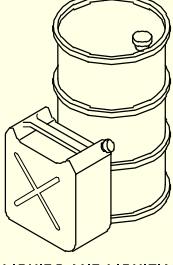
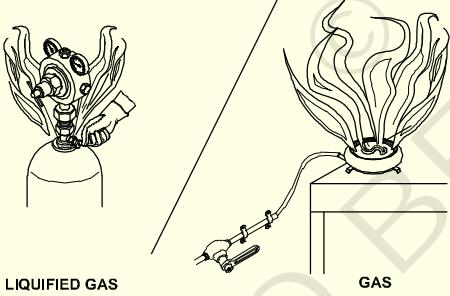
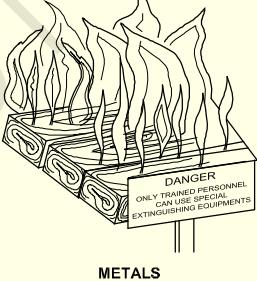
Blowlamps and torches must not be left burning when they are not in use.

Extinguishing fires: Fires are classified into four types in terms of the nature of fuel.

Different types of fires (Fig 2, Fig 3 Fig 4 & Fig 5) have to be dealt with in different ways and with different extinguishing agents.

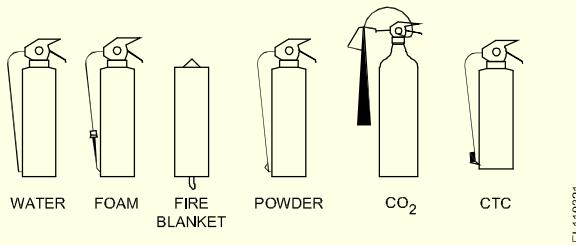
An extinguishing agent is the material or substance used to put out the fire, and is usually (but not always) contained in a fire extinguisher with a release mechanism for spraying into the fire.

It is important to know the right type of agent for extinguishing a particular type of fire; using a wrong agent can make things worse. There is no classification for 'electrical fires' as such, since these are only fires in materials where electricity is present.

Fuel	Extinguishing
<p>Fig 2</p> <p style="text-align: center;">CLASS 'A' FIRE</p>  <p style="text-align: right;">EL110312</p>	<p>Most effective ie. cooling with water. Jets of water should be sprayed on the base of the fire and then gradually upwards.</p>
<p>Fig 3</p> <p style="text-align: center;">CLASS 'B' FIRE</p>  <p style="text-align: right;">EL110313</p>	<p>Should be smothered. The aim is to cover the entire surface of the burning liquid. This has the effect of cutting off the supply of oxygen to the fire.</p> <p>Water should never be used on burning liquids.</p> <p>Foam, dry powder or CO₂ may be used on this type of fire.</p>
<p>Fig 4</p> <p style="text-align: center;">CLASS 'C' FIRE</p>  <p style="text-align: right;">EL110314</p>	<p>Extreme caution is necessary in dealing with liquefied gases. There is a risk of explosion and sudden outbreak of fire in the entire vicinity. If an appliance fed from a cylinder catches fire - shut off the supply of gas. The safest course is to raise an alarm and leave the fire to be dealt with by trained personnel.</p> <p>Dry powder extinguishers are used on this type of fire.</p>
<p>Fig 5</p> <p style="text-align: center;">CLASS 'D' FIRE</p>  <p style="text-align: right;">EL110315</p>	<p>Special powders have now been developed which are capable of controlling and/or extinguishing this type of fire.</p> <p>The standard range of fire extinguishing agents is inadequate or dangerous when dealing with metal fires.</p> <p>Fire on electrical equipment.</p> <p>Halon, Carbon dioxide, dry powder and vapourising liquid (CTC) extinguishers can be used to deal with fires in electrical equipment. Foam or liquid (eg. water) extinguishers must not be used on electrical equipment under any circumstances.</p>

Many types of fire extinguishers are available with different extinguishing 'agents' to deal with different classes of fires. (Fig 1)

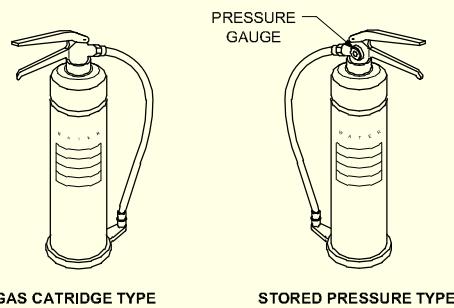
Fig 1



EL110321

Water-filled extinguishers: There are two methods of operation. (Fig 2)

Fig 2



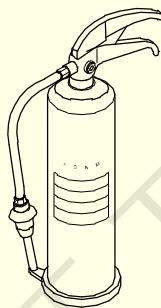
EL110322

- Gas cartridge type
- Stored pressure type

With both methods of operation the discharge can be interrupted as required, conserving the contents and preventing unnecessary water damage.

Foam extinguishers (Fig 3): These may be of stored pressure or gas cartridge types. Always check the operating instructions on the extinguisher before use.

Fig 3



EL110323

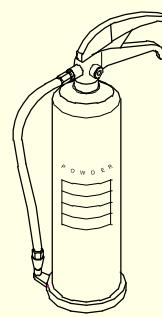
Most suitable for

- flammable liquid fires
- running liquid fires.

Must not be used on fires where electrical equipment is involved.

Dry powder extinguishers (Fig 4): Extinguishers fitted with dry powder may be of the gas cartridge or stored pressure type. Appearance and method of operation is the same as that of the water-filled one. The main distinguishing feature is the fork shaped nozzle. Powders have been developed to deal with class D fires.

Fig 4



EL110324

Carbon dioxide (CO₂): This type is easily distinguished by the distinctively shaped discharge horn. (Fig 5).

Fig 5



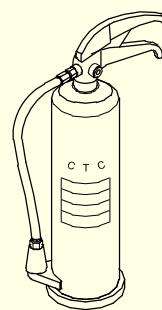
EL110325

Suitable for Class B fires. Best suited where contamination by deposits must be avoided. Not generally effective in open air.

Always check the operating instructions on the container before use. Available with different gadgets of operation such as - plunger, lever, trigger etc.

Halon extinguishers (Fig 6): These extinguishers may be filled with carbon-tetrachloride and Bromochlorodifluoro methene (BCF). They may be either gas cartridge or stored pressure type.

Fig 6



EL110326

They are more effective in extinguishing small fires involving pouring liquids. These extinguishers are particularly suitable and safe to use on electrical equipment as the chemicals are electrically non-conductive.

The fumes given off by these extinguishers are dangerous, especially in confined space.

The general procedure in the event of a fire:

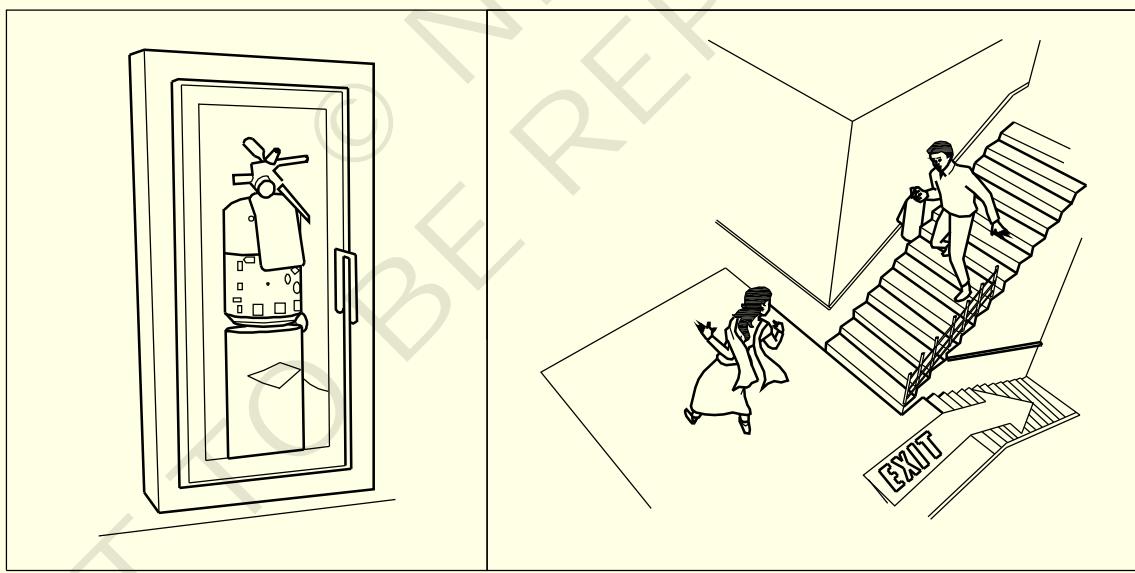
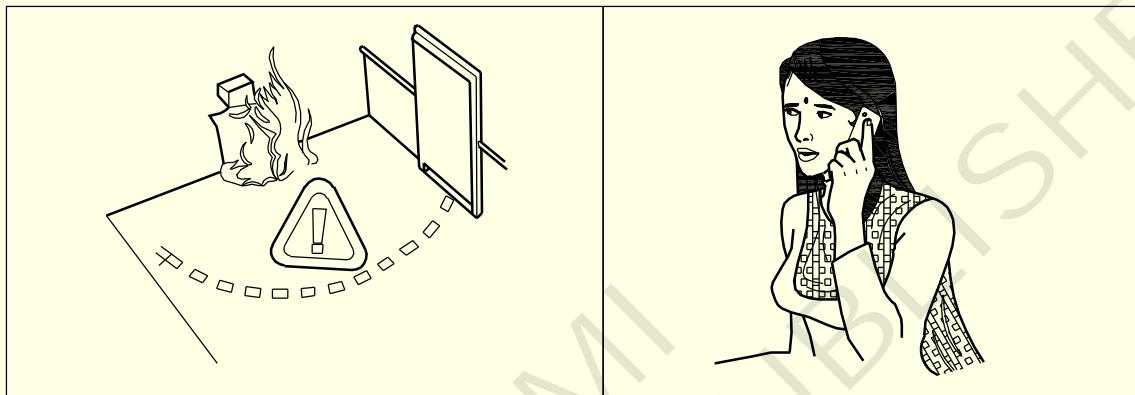
- Raise an alarm.
- Turn off all machinery and power (gas and electricity).

- Close the doors and windows, but do not lock or bolt them. This will limit the oxygen fed to the fire and prevent its spreading.
- Try to deal with the fire if you can do so safely. Do not risk getting trapped.
- Anybody not involved in fighting the fire should leave calmly using the emergency exits and go to the designated assembly point. Failure to do this may mean that some person being unaccounted for and others may have to put themselves to the trouble of searching for him or her at risk to themselves.

Practice on fire extinguishers

Objectives: At the end of this lesson you shall be able to

- state about the selection of the fire extinguishers according to the type of fire
- state the method of operation of the fire extinguisher
- explain how to extinguish the fire.



FI1104E1

PROCEDURE (Fig 1)

- Alert people surrounding by shouting fire, fire, fire when observe the fire.
- Inform fire service or arrange to inform immediately.
- Open emergency exist and ask them to go away.
- Put "off" electrical power supply.

Don't allow people to go nearer to the fire.

- Analyze and identify the type of fire. Refer Table1.

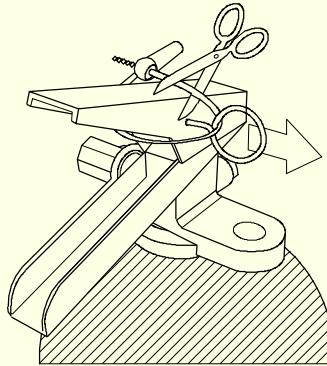
Table-1

Class 'A'	Wood, paper, cloth, solid material
Class 'B'	Oil based fire (grease, gasoline, oil) liquefiable gases
Class 'C'	Gas and liquefiable gases
Class 'D'	Metals and electrical equipment

Assume the fire is 'B' type (flammable liquifiable solids)

- Select CO₂ (Carbon dioxide) fire extinguisher.
- Locate and pickup, CO₂ fire extinguisher. Click for its expiry date.
- Break the seal (Fig 2)

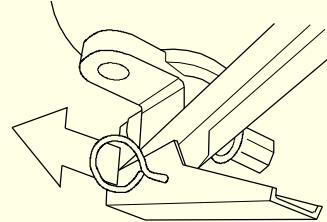
Fig 2



F1104H2

- Pull the safety pin from the handle (Pin located at the top of the fire extinguisher) (Fig 3)

Fig 3

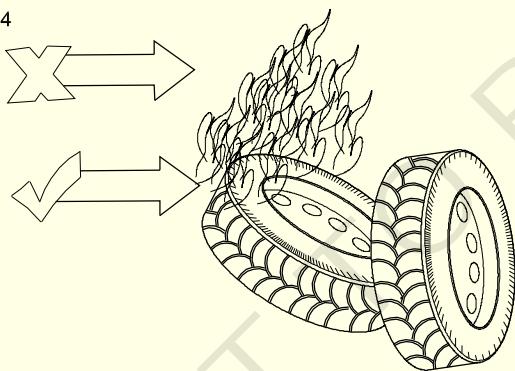


F1104H3

- Aim the extinguisher nozzle or hose at the base of the fire (this will remove the source of fuel fire) (Fig 4)

Keep your self low

Fig 4

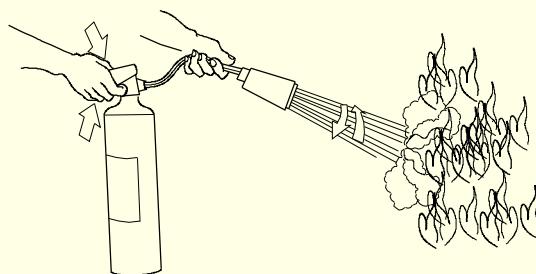


F1104H4

- Squeeze the handle lever slowly to discharge the agent (Fig 5)

- Sweep side to side approximately 15 cm over the fuel fire until the fire is put off (Fig 5)

Fig 5



F1104H5

Fire extinguishers are manufactured for use from the distance.

Caution

- While putting off fire, the fire may flare up
- Do not be panick belong as it put off promptly.
- If the fire doesn't respond well after you have used up the fire extinguisher move away yourself away from the fire point.
- Do not attempt to put out a fire where it is emitting toxic smoke leave it for the professionals.
- Remember that your life is more important than property. So don't place yourself or others at risk.

In order to remember the simple operation of the extinguisher. Remember P.A.S.S. This will help you to use the fire extinguisher.

P for Pull

A for Aim

S for Squeeze

S for Sweep

Introduction to computers

Objectives : At the end of this lesson you shall be able to

- define and classify computers
- list the advantages and limitations of computers
- list the applications of computer
- describe the voltages and currents in the computer.

Computer - Definitions: A computer is an electronic machine, operating under the control of instructions stored in its own memory that can accept data (input), manipulate the data according to specified rules (process), produce results (output), and store the results for future use.

Technically, a computer is a programmable machine. This means it can execute a list of programmed instructions and respond to new instructions that it is given.

History of Computer

Charles babbage's machine

The working principles of today's computers were provided by an English mathematician Charles Babbage around 1833's invented a machine called the "Analytical Engine". A machine which could calculate and print tables of functions using limited techniques.

The Analytical Engine had four parts. A mill, which was the section which did the calculations, essentially the CPU; the store, where the information was kept recorded, essentially the memory; the reader, which would allow data to be entered using punched cards, essentially the keyboard, and the printer.

Hence, Charles Babbage is considered as the "Father of the Computer" as in Fig 1.

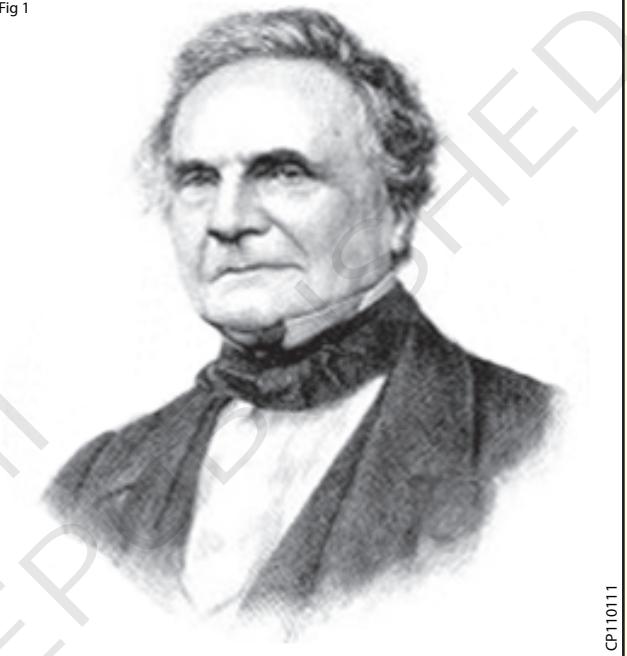
The generations of computers are characterized by a major technological development that fundamentally changed the way computers operate, resulting in increasingly smaller, cheaper, more powerful and more efficient and reliable devices. The various generations of computers are listed below:

First Generation (1946-1954): In 1946 the digital computer using **electronic valves** (Vacuum tubes) are known as first generation computers. The first '**computer**' to use electronic valves i.e. vacuum tubes. The high cost of vacuum tubes prevented their use for main memory. They stored information in the form of propagating sound waves.

The vacuum tube consumes a lot of power. These computers were large in size and writing programs on them was difficult. Some of the computers of this generation were:

Mark I :The IBM Automatic Sequence Controlled Calculator (ASCC), called the Mark I by **Harvard University**, was an **electro-mechanical computer**. Mark I is the first machine to successfully perform a long series of **arithmetic and logical operation**. Mark I is the **First Generation Computer**.

Fig 1



CP10111

ENIAC: It was the first electronic computer built in 1946 at University of Pennsylvania, USA by John Eckert and John Mauchy. It was named Electronic Numerical Integrator and Calculator (ENIAC). The ENIAC was 30-50 feet long, weighted 30 tons, contained 18,000 vacuum tubes, 70,000 resistors, 10,000 capacitors and required 150,000 watts of electricity. Today computer is many times as powerful as ENIAC, still size is very small.

EDVAC: It stands for **Electronic Discrete Variable Automatic Computer** and was developed in 1950. The concept of storing data and instructions inside the computer was introduced here. This allowed much faster operation since the computer had rapid access to both data and instructions. The other advantage of storing instruction was that computer could do logical decision internally. The EDVAC was a **binary serial computer** with automatic addition, subtraction, multiplication, programmed division and automatic checking with an ultrasonic serial memory.

EDSAC: It stands for **Electronic Delay Storage Automatic Computer** and was developed by **M.V. Wilkes at Cambridge University in 1949**. The EDSAC is the first **stored-program computer**. The EDSAC performed computations in the three millisecond range. It performed arithmetic and logical operations without human intervention. The key to the success was in the stored instructions which it depended upon solely for its operation.

This machine marked the beginning of the computer age.

UNIVAC-1: It stands for Universal Automatic computer and it was the First commercial computer developed by United States In 1951. The machine was 25 feet by 50 feet in length, contained 5,600 tubes, 18,000 crystal diodes, and 300 relays. It utilized serial circuitry, 2.25 MHz bit rate, and had an internal storage capacity 1,000 words or 12,000 characters.

The UNIVAC was used for **general purpose computing** with large amounts of input and output. The UNIVAC was also the first computer to come equipped with a magnetic tape unit and was the **first computer to use buffer memory**.

Limitations of First Generation Computer

Followings are the major drawbacks of First generation computers.

- They used valves or vacuum tubes as their main electronic component.
- They were large in size, slow in processing and had less storage capacity.
- They consumed lots of electricity and produced lots of heat.
- Their computing capabilities were limited.
- They were not so accurate and reliable.
- They used machine level language for programming.
- They were very expensive.

Second Generation (1955-1964): The second-generation computer used **transistors** for CPU components and **ferrite cores for main memory&magnetic disks** for secondary memory. They used high-level languages such as **FORTRAN (1956), ALGOL (1960) & COBOL (1960 - 1961)**. Input Output (I/O)processor was included to control I/O operations.

Manufacturing cost was also very low. Thus the size of the computer got reduced considerably.

It is in the second generation that the concept of Central Processing Unit (CPU), memory, programming language and input and output units were developed. Some of the second generation computers are IBM 1620, IBM 1401,CDC 3600.

Features

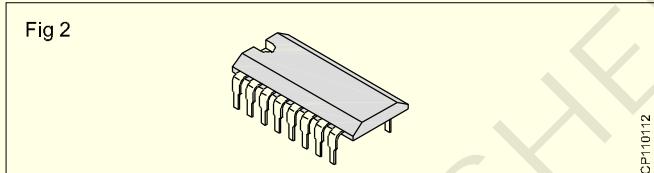
- Transistors were used instead of Vacuum Tube.
- Processing speed is faster than First Generation Computers (Micro Second)
- Smaller in Size (51 square feet)
- The input and output devices were faster.

Third Generation (1964-1977): By the development of a small chip consisting of the capacity of the **300 transistors**. These Integrated Circuits (IC)s are popularly known as **Chips**.

A single IC has many transistors, registers and capacitors built on a single thin slice of silicon. So it is quite obvious that the size of the computer got further reduced. Some of the computers developed during this period were **IBM-360, ICL-1900, IBM-370, and VAX-750**. Higher level language such as **BASIC (Beginners All purpose Symbolic Instruction Code)** was developed during this period.

Computers of this generation were small in size, low cost, large memory and processing speed is very high. Very soon ICs were replaced by **LSI (Large Scale Integration)**, which consisted about 100 components.

An IC containing about 100 components is called LSI as in (Fig 2).



Features

- They used Integrated Circuit (IC) chips in place of the transistors.
- Semiconductor memory devices were used.
- The size was greatly reduced, the speed of processing was high, and they were more accurate and reliable.
- Large Scale Integration (LSI) and Very Large Scale Integration (VLSI) were also developed.
- The mini computers were introduced in this generation.
- They used high level language for programming.

Fourth Generation (1978 - present): An IC containing about 100 components is called LSI (Large Scale Integration) and the one, which has more than 1000 such components, is called as VLSI (Very Large Scale Integration).

It uses large scale Integrated Circuits(LSIC) built on a single silicon chip called microprocessors. Due to the development of microprocessor it is possible to place computer's central processing unit(CPU) on single chip. These computers are called microcomputers.

Later very large scale Integrated Circuits(VLSIC) replaced LSICs. Thus the computer which was occupying a very large room in earlier days can now be placed on a table. The personal computer (PC) that you see in your school is a Fourth Generation Computer Main memory used fast semiconductors chips up to 4 M bits size. Some of the Fourth generation computers are IBM PC, Apple-Macintosh, etc.

Hard disks were used as secondary memory. Keyboards, dot matrix printers etc. were developed. Operating System (OS)-such as MS-DOS, UNIX, Apple's Macintosh were available. Object oriented language, C++ etc were developed.

Features

- They used Microprocessor (VLSI) as their main switching element.
- They are also called as micro computers or personal computers.
- Their size varies from desktop to laptop or palmtop.
- They have very high speed of processing; they are 100% accurate, reliable, diligent and versatile.
- They have very large storage capacity.

Fifth Generation (PRESENT AND FUTURE): 5th generation computers use ULSI (Ultra-Large Scale Integration) chips. Millions of transistors are placed in a single IC in ULSI chips.

64 bit microprocessors have been developed during this period.. Memory chips and flash memory up to 1 GB, hard disks up to 600 GB & optical disks up to 50 GB have been developed. (Fig 3)

Fifth generation computing devices, based on Artificial Intelligence, are still in development, though there are some applications, such as voice recognition, that are being used today.

Artificial Intelligence is the branch of computer science concerned with making computers behave like humans. The term was coined in 1956 by John McCarthy at the Massachusetts Institute of Technology. Artificial intelligence includes:

Fig 3



- **Games Playing:** Programming computers to play games such as chess and checkers
- **Expert Systems:** Programming computers to make decisions in real-life situations (for example, some expert systems help doctors diagnose diseases based on symptoms)
- **Natural Language:** Programming computers to understand natural human languages
- **Neural Networks:** Systems that simulate intelligence by attempting to reproduce the types of physical connections that occur in animal brains
- **Robotics:** programming computers to see and hear and react to other sensory stimuli

Table - 1

Generation	Electronic component	Advantages	Disadvantages
First	Vaccum tube and computational work	Helped in calculation	1 Big size 2 Very costly 3 Slow speed 4 Low accuracy 5 Low storage 6 High power requirements 7 High heat generation 8 High failure rate 9 Used machine language 10 No operating system
Second	Transistor	1 Smaller size 2 Less cost 3 Better speed 4 Low power consumption and less heat generation 5 Better storage capacity 6 Better accuracy and more reliability	1 Need air conditioning 2 Constant maintenance 3 No operating systems 4 Later stage computers used assembly languages
Third	Integrated Circuits(IC) small & medium scale	1 Better in all aspects compared to I & II 2 Used operating systems and high level language	1 Initial problem with manufacturers 2 No insight obtained into internal working
Fourth	VLSI or Microprocessor	1 Low cost 2 Excellent speed and reliability 3 Computers close to man	1 Less powerful than main frame computers

Fifth (Knowledge Information Processing Systems)	ULSI or Bio-Chips	1 Very cheap 2 Super speeds 3 Very high storage capacity 4 Highly sophisticated OS 5 Posses intelligence and decision making ability	1 New low level language needed
--	-------------------	--	---------------------------------

Classification of computers

Computers are classified according to the following criteria:

- Principle of Operation
- Computing Power, Memory Capacity and cost
- Technological Development
- Principle of operation
 - a Analog computer
 - b Digital Computer
 - c Hybrid Computer

Analog Computer

It is a computer that measures continuously changing physical quantities such as current, temperature, pressure etc. and converts them into quantities which can be used as data for computation. As these computers deal with continuously varying quantities they will give only approximate results. Its output is usually displayed on a meter or scale. Analog computer has low memory and fewer functions. These are used for engineering and scientific applications.

- Thermometer
- Speedometer
- Analog clock

Digital Computer

A digital computer works with digital data. Digital computer uses binary number system. Binary number system consists of only two digits '0' and '1'. A digital computer represents data in digital signals. A '0' represents OFF and a '1' represents ON. Digital computer performs arithmetic and logical operations on data. It gives output in digital form.

Digital computers are very fast. These computers can store results. They have large Memory (that is data storing capacity). Today most of the computers used in offices and homes are Digital computers.

The digital computers are further divided into the following two groups:

- Special purpose computers
- General purpose computers

Table 2

Analog Computers	Digital Computers
<p>1 Analog Computers Work on continuous values.</p> <p>2 Analog Computers have low memory.</p> <p>3 Analog computers have Slow speed.</p> <p>4 Analog computers are less reliable.</p> <p>5 Analog computers used in engineering</p> <p>6 Analog computers are used to calculate / measure analog quantities like speed and temperature.</p> <p>7 Analog computers provide less accurate results.</p> <p>8 Normally Analog Computers are specific purpose</p> <p>9 Normally Analog Computers are specific purpose</p> <p>10 Examples of Analog computers are: thermometer, analog clock, speedometer etc.</p>	<p>Digital computers Work on discrete values.</p> <p>Digital computers have a very large memory</p> <p>Digital computers have fast speed.</p> <p>Digital computers are more reliable.</p> <p>Digital computers are used in all fields of life, science and medical fields.</p> <p>Digital computers are used to calculate mathematical and logical operations. It can solve addition, subtraction, division, multiplication and other mathematical and statistical operations.</p> <p>Digital computers provide 100% accurate results.</p> <p>Digital Computers are general purpose</p> <p>Digital Computers are general purpose</p> <p>Examples of digital computers are: Personal Computer, laptops, smart phones etc.</p>

Special Purpose Computers

It is a computer designed to solve specific type of problem. The computers used in ships and aircrafts, etc.

General Purpose computers

It is a computer designed to solve a wide variety of problems. A General purpose Computer can store different programs and process them.

The differences between analog and digital computers are listed in table 2

Hybrid Computer (Fig 4)

A hybrid computer is a combination of both analog and digital computer. Hybrid computer can handle both analog and digital data. A hybrid computer combines the best characteristics of both the analog and digital computer. It can accept data in both analog and digital form.

Fig 4



CPI10114

Applications

Hybrid computer devices are used in hospitals that may calculate patient's heart function, temperature and blood pressure etc. This calculation may be converted into numbers and shown in digital form. For example, The Vital Signs Monitoring unit also called (VSM) in short. It has Blood Pressure monitor, ECG monitor, respiratory monitor, and is also used for monitoring anesthesia.

- Hybrid computers are also used in spaceships and missile system.
- Hybrid Computer Machines are generally used in scientific applications
- Hybrid computers are used for controlling industrial processes.

Computers are classified on the basis of computing power, memory capacity and cost.

- Microcomputer or Personal Computer.
- Mini Computer.
- Mainframe Computer.
- Super Computer.

Micro Computers

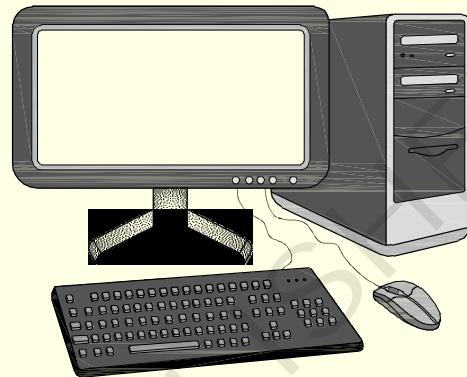
Micro computer is also called personal computer. It was introduced in 1970. Examples of personal computers are

PC and Apple Macintosh. The major types of personal computers are desktop computer and portable computer.

Desktop Computer

These computers can easily fit on a table or desktop, hence the name. These computers come in two models or casings. In Desktop model, the system unit is placed on the desktop or table. Monitor is placed on the system unit. In Tower model, both monitor and system unit are placed on the table as in (Fig 5).

Fig 5



TOWER MODEL COMPUTER

CPI10115

Portable computer: Portable is a personal computer that can be carried from one place to other easily. Notebook computer and handheld computer (smart phone) are examples of portable computers. Notebook computer is also called laptop computer. Laptop computers are very small in size and can be placed easily on lap.

Lap top computer or notebook computer: The laptop computer or notebook computer will be as shown in fig 6a and fig 6b. It is also called as tower model computer.

Palmtop Computer/Digital Diary /Notebook /PDAs

A handheld computer (like smart phone) is also portable. Hand held computer is known as palmtop computer. Palmtops have no keyboard but the screen serves both as an input and output device. It easily fits in the hand of the user.

Fig 6a



LAP TOP COMPUTER OR NOTE BOOK COMPUTER

CPI10116A

Fig 6



Uses of Micro Computer

The PC is the most common type of computer used in the office. It is now widely used in many homes. These are also used for business and engineering application.

Mini Computer

Mini computers were introduced in the 1960s. Minicomputer is larger and more powerful computer than personal computer. It can execute five million instructions per second. It generally consists of two or more processors.

Minicomputer can serve up to 4000 connected users simultaneously. It is normally accessed by users via personal computer or terminal. A device with a monitor and keyboard is called terminal. It is also known as dumb terminal. It has no processing power and cannot work as stand-alone computer. Some of the minicomputers models are VAX-8800, AS400

Uses of Mini Computer

Mini computers are often used by small and medium-sized companies to provide centralized store of information.

Mainframe Computer

Mainframe computers were introduced in 1975. A mainframe computer is a very large computer in size. It is processors. It is designed to perform multiple tasks for multiple users at the same time. Mainframe computers can serve up to 50,000 users at the same time.

The users access a mainframe computer through terminal or personal computer. A typical mainframe computer can execute 16 million instructions per second. Some of the main computers models are

- NEC 610
- DEC 10

Uses of Mainframe Computer

Mainframe computers are used primarily by corporate and governmental organizations for critical applications, bulk data processing such as census, industry and consumer statistics, and transaction processing.

Super computer

Super computers were introduced in 1980s. Super computer is the fastest computer. Super computer is the biggest in size and the most expensive in price than any other computers.

It is the most sophisticated, complex and advanced computer. It has a very large storage capacity. It can process trillions of instructions in one second. Super Computer is the fastest and most powerful computer of a time. Supercomputers are very expensive. Supercomputers are used for highly calculation-intensive tasks. Super computers are also used for specialized applications that require immense amounts of mathematical calculations.

Applications of Super Computer

- Weather forecasting,
- Animated graphics like in Hollywood movies,
- Fluid dynamic calculations
- Nuclear energy research
- Space science
- Weapon and missile design
- Petroleum exploration, and etc.

Today, supercomputers are produced by traditional companies such as Cray, IBM and Hewlett-Packard. Since October 2010, the Tianhe-1A supercomputer has been the fastest in the world; it is located in China.

The main difference between a supercomputer and a mainframe is that a supercomputer channels all its power into executing a single program as fast as possible, whereas a mainframe uses its power to execute many programs concurrently. The modern super computer consists of thousands of microprocessors. Super computer uses high-speed facilities such as satellite for online processing.

Sum of the super computers models are CRAY-XP, ETA-10, Param and Deep Blue .

Advantages of computers

- A computer has a very high processing speed with high reliability.
- Large volume of information can be stored in the memory any particular data/program can be retrieved immediately.
- Solution to a complicated problem is possible at a very high speed.
- Processing of large volume of data saves a lot of clerical work which reduces the processing cost.
- Computers perform operations efficiently at environments where presence of human being is not possible such as furnace, poisonous atmosphere, vacuum, unmanned satellite, etc.

Limitation of computers

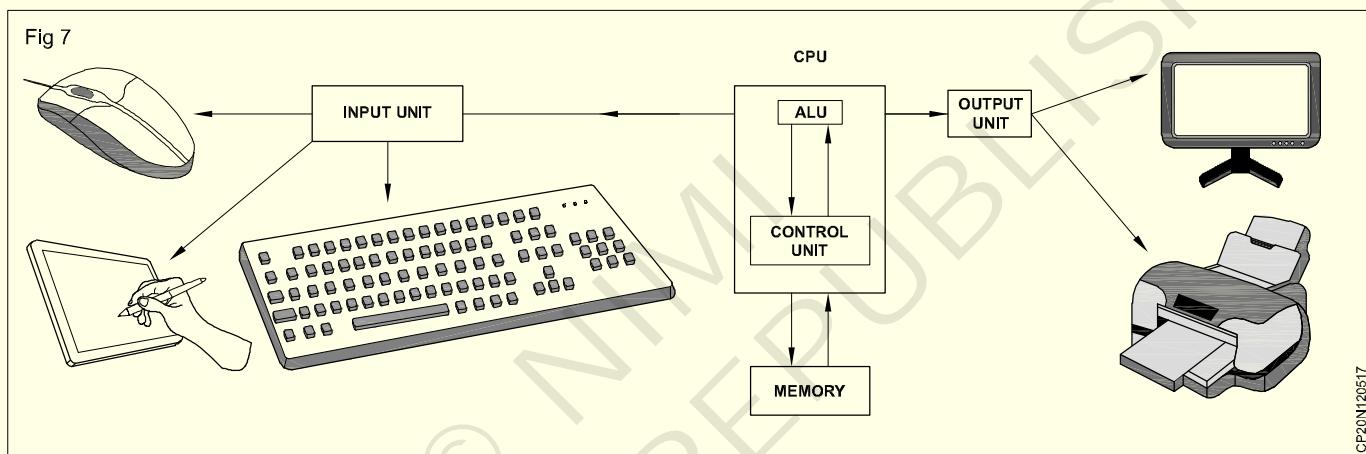
- High initial cost.

- Input information has to be prepared in the form of statements called program which requires a considerable amount of knowledge.
- usage of computers will be economical only when there is clerical data processing for large volume of data and are repetitive in nature
- It is a merely a machine it cannot correct errors on its own.

Functions of Computers

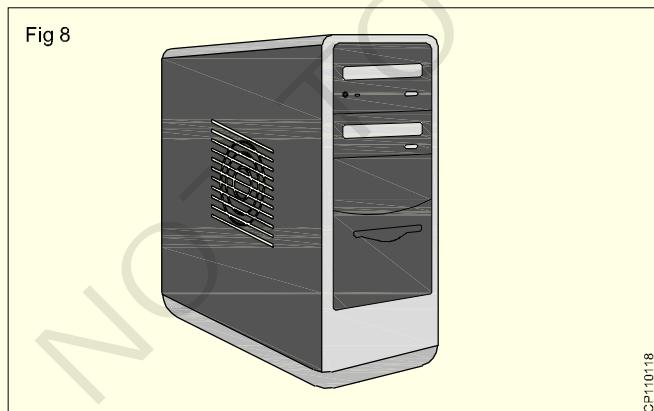
All computers are made up of following basic units as shown in fig (7). They are as follows:-

- 1 Input Unit
- 2 Central processing Unit (CPU)
 - a Arithmetic Logic Unit(ALU)
 - b Control Unit (CU)



Central Processing Unit (CPU) (Fig 8)

The central processing unit (CPU) is the electronic brain of the computer as in fig-8. The CPU in a personal computer is usually a single chip. It organizes and carries out instructions that come from either the user or from the software. The processor is made up of many components. CPU performs the following functions:



- It performs all calculations.
- It takes all decisions.
- It controls all units of the computer.

Two typical components of a **CPU** are the following:

- 3 Memory
- 4 Output Unit

Input Unit

Computers need to receive data and instruction in order to solve any problem. Therefore we need to input the data and instructions into the computers. The input unit consists of one or more input devices. Keyboard is the one of the most commonly used input device. Some of the input devices are listed in table 1.

Input devices perform the following functions.

- Accept the data and instructions from the outside world.
- Convert it to a form that the computer can understand.
- Supply the converted data to the computer system for further processing.

The arithmetic logic unit (ALU), which performs arithmetic and logical operations.

The control unit (CU), which extracts instructions from memory and decodes and executes them, calling on the ALU when necessary.

Memory

Memory refers to the physical device used to store the program or data on the temporary or permanent basis for use in a computer or other digital electronic device.

There are two types of memory in computer.

- Primary Memory
- Secondary Memory

Output Unit

Output unit receive the informations from the processing unit and provide the results in human readable form.

Output Devices

The some of the output devices are

- Monitor
- Printer
- Plotter
- Speaker

Applications of computers

Science: Scientists have been using computers to develop theories and to analyse and test the data. The high speed and accuracy of the computer allow different scientific analyses to be carried out. They can be used to generate detailed studies of how earthquakes affect buildings or pollution affects weather pattern. Satellite-based applications have not been possible without the use of computers. Moreover, it would not be possible to get the information of the solar system and the cosmos without computers.

Education: Computers have also revolutionized the whole process of education. Currently, the classrooms, libraries and museums are efficiently utilizing computers to make the education much more interesting. Unlike recorded television shows, computer-aided education (CAE) and computer-based training (CBT) packages are making learning much more interactive.

Medicine and Health Care: There has been an increasing use of computers in the field of medicine. Now, doctors are using computers right from diagnosing the illness to monitoring a patient's status during complex surgery. By using automated imaging techniques, doctors are able to look inside a person's body and can study each organ in detail (e.g. CT scans or MRI scans), which was not possible few years ago. There are several examples of special-purpose computers that can operate within the human body such as cochlear implant, a special kind of hearing aid that makes it possible for deaf people to hear.

Engineering/Architecture/Manufacturing: The architects and engineers are extensively using computers in designing and drawings. Computers can create objects that can be viewed from all the three dimensions. By using techniques like virtual reality, architects can explore houses that have been designed but not built. The manufacturing factories are using computerized robotic arms to perform hazardous jobs. Besides, computer-aided manufacturing (CAM) can be used in designing the product, ordering the parts and planning production. Thus, computers help in coordinating the entire manufacturing process.

Entertainment: Computers are finding greater use in entertainment industry. They are used to control the images and sounds. The special effects, which mesmerize the audience, would not have been possible without the computers. In addition, computerized animation and colourful graphics have modernized the film industry.

Communication: E-mail or electronic mail is one of the communication media in which computer is used. Through e-mail, messages and reports are passed from one person to one or more persons with the aid of computer and telephone line. The advantage of this service is that while transferring the messages it saves time, avoids wastage of paper and so on. Moreover, the person who is receiving the messages can read the messages whenever he is free and can save it, reply it, forward it or delete it from the computer.

Business Application: This is one of the important uses of the computer. Initially, computers were used for batch-

processing jobs, where one does not require the immediate response from the computer. Currently, computers are mainly used for real-time applications (like at the sales counter) that require immediate response from the computer. There are various concerns where computers are used such as in business forecasting, to prepare pay bills and personal records, in banking operations and data storage, in various types of life insurance business and as an aid to management. Businesses are also using the networking of computers, where a number of computers are connected together to share the data and the information. Use of e-mail and the Internet has changed the ways of doing business.

Publishing: Computers have created a field known as desktop publishing (DTP). In DTP, with the help of computer and a laser printer one can perform the publishing job all by oneself. Many of the tasks requiring long manual hours such as making table of contents and index can be automatically performed using the computers and DTP software.

Banking: Computers are extensively used in the field of banking and finance. People can use the ATM (automated teller machine) services 24 hours a day to deposit and withdraw cash. When different branches of the bank are connected through computer networks, the inter branch transactions such as cheque and draft can be performed without any delay.

Railway Reservation System

Using this system, the user can perform following operations through online. (web site: www.irctc.co.in)

- search the train and its timings
- check seats and birth availability
- booking and cancelling tickets
- status of PNR (Passenger Name Record)

Telephone / Electricity Board Billing:

The users can do the following operations through online by using this system. (Web site: portal.bsnl.in - BSNL)

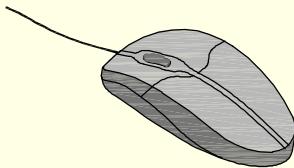
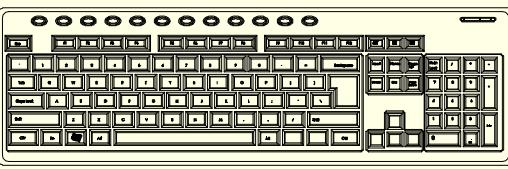
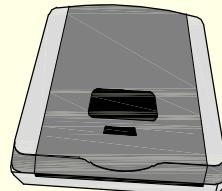
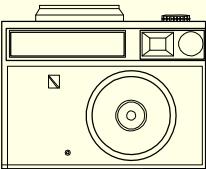
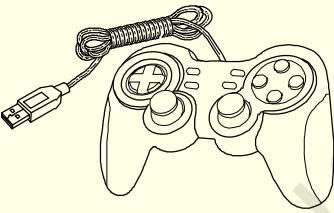
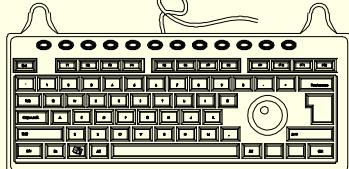
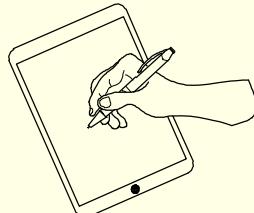
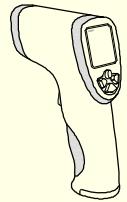
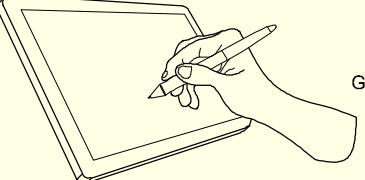
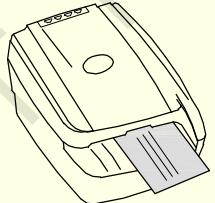
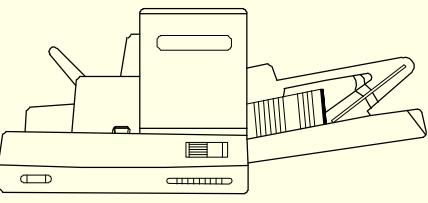
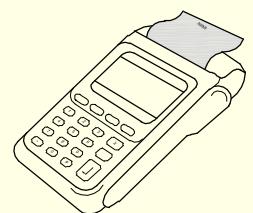
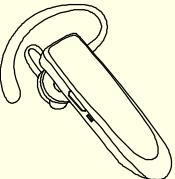
- Register the telephone / electricity board number
- Check and pay the bill amount
- Register the complaints

E-Governance

E-Governance implies technology driven governance. E-Governance is the application of Information and Communication Technology (ICT) for delivering government services, exchange of information communication transactions, integration of various stand-alone systems and services between Government-to-Citizens (G2C), Government-to-Business(G2B),Government-to-Government (G2G) as well as back office processes and interactions within the entire government frame work.

E-Governance covers all the sectors with a view to providing hassle free, transparent and efficient service to the common man (both in urban and rural areas).

Fig 9

 MOUSE	 KEYBOARD
 SCANNER	 CAMERA
 WEB CAMERA	 JOY STICK
 TARCK BALL	 TOUCH PAD
 LIGHT PEN	 BAR CODE READER
 MICROPHONE	 GRAPHIC TABLET
 MAGNETIC INK CHARACTER READER	 OPTICAL MARK READER
 MAGNETIC CARD READER	 BLUETOOTH

Introduction to CPU architecture and motherboard

Objectives: At the end of this lesson you shall be able to

- state what is hardware and test the internal and external hardware
- brief the listed hardware
- state what is partitions and their types
- explain the booting and its procedures.

Computer Hardware

The physical units of a computer are called as the hardware of a computer.

Internal hardware examples

- Blu-Ray, CD-ROM, and DVD
- CPU
- Hard drive
- Motherboard
- RAM
- Sound card
- Video card
- SMPS

External hardware examples

- Flat-panel, Monitor, and LCD
- Keyboard
- Mouse
- Printer
- Scanner

CPU & ALU

Central Processing Unit (CPU) is the heart of the Computer. It is the hardware that carries out the instructions of a computer program by performing the basic arithmetical, logical, and input/output operations of the system.

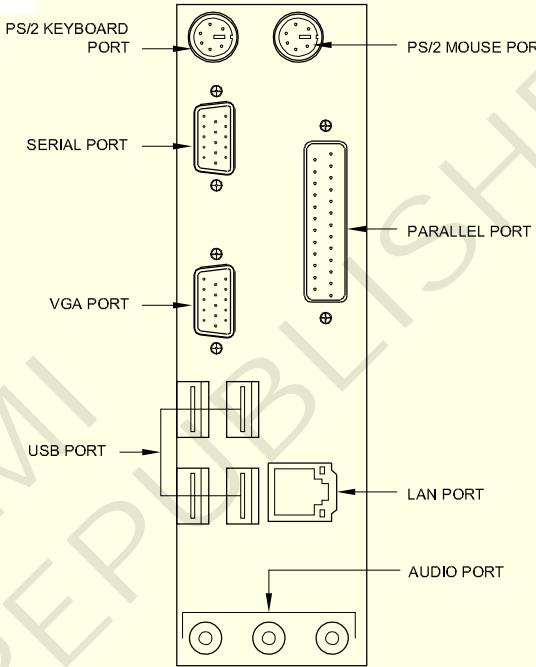
CPU Ports and Connectors

A port is a connector at the back of a Computer cabinet where you plug in an external device such as a printer, keyboard, scanner, modem etc. This allows instructions and data to flow between the computer and the device. The computer ports are also commonly referred to as the Input/output ports (I/O ports). These ports can be either serial or parallel. Fig 1 shows the commonly available ports on a personal computer.

Most connectors are separated, permitting the cable to be plugged in only in the correct direction. The keyboard and mouse use "PS2" (Personal System 2) connectors. The PS2 connectors are color-coded. The purple connector is for the keyboard. The green connector is for the mouse.

- **PS/2 Ports:** Standard keyboards and mouse often connect to the computer via the PS/2 ports. To plug in

Fig 1



CP110119

a keyboard or mouse cable, first match the cable to the connector. Then push the cable into the connector. Be sure not to force the connector because you will end up bending the pins

- **Serial & Parallel Ports:** The serial port and parallel port allow connections to printers and other external devices. To transfer a byte through a serial port, eight bits are queued and sent bit by bit. However, in a parallel port, all the eight bits are transferred simultaneously
- The parallel port, serial port, and video port all use "D" type connectors (DB-25M, DB-9M and DB-15F respectively). These are called D connectors because of their shape, which permits the cables to be plugged in only one way.

USB (Universal serial bus) Ports

Devices like digital cameras, scanners and printers often connect to the motherboard via the USB ports. A USB connector's distinctive rectangular shape makes it easily recognizable.

USB has a number of features that makes it particularly popular on PCs. First, USB devices are hot swappable. You can insert or remove them without restarting your system

LAN(Local Area Network) Port: The LAN port is used to connect the PC to a local network or to high speed Internet services.

VGA (Video Graphics Array) Ports: The VGA port provides access to integrated video.

Audio ports: It provides access to integrated audio.

The audio jacks are the most confusing connectors on the back panel. Although the jacks are sometimes color-coded, the devices that plug into them rarely.

CPU Front Panel (Fig 2)



It may contain the following parts.

- Power On/Off Switch
- Power Indicator
- CD/DVD Drive
- CD/DVD Drive Open/Close Button
- CD/DVD Drive indicator
- Floppy Disk Drive
- Floppy Disk Drive Indicator
- USB Ports
- Audio and Mic connectors

Design of CPU Cabinets may vary based on the manufacturer.

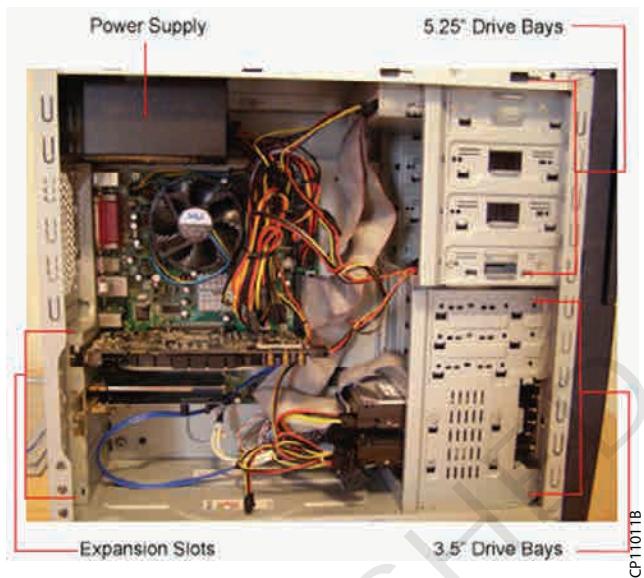
The System Unit and Its Components

The system unit is a box-like unit filled with a number of useful components, each performing a discrete function. These components work together to accomplish the main function of the computer, viz. accept and process input and deliver output. This section will elaborate on these components one by one. Fig 3 shows the various components of the system unit.

Power Supply

The power supply connects to nearly every device in the PC to provide power. It is located at the rear of the case. The system unit draws power from the AC mains through a power protection device.

Fig 3



This power is not directly supplied to the internal components. Instead, one of the components, called the internal power supply, converts the AC input into DC output of 5 and 12 volts. Normally, the **internal power supply** is referred to as **Switched Mode Power Supply (SMPS)**.

The SMPS provides cable connectors to supply the required voltage to the other internal components like the floppy drives, the hard disk drive, the motherboard and external device such as the keyboard. The ON/OFF switch of the system unit is actually a part of the SMPS.

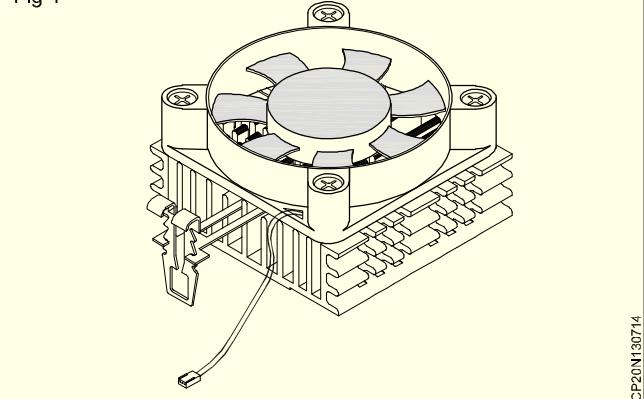
Fan

The **SMPS** has a small fan, called the exhaust fan, attached to **SMPS (Fig 4)**. This fan rotates as long as the computer is switched on. Its function is to cool the **SMPS** unit.

Drive Bays: The 5.25" and 3.5" drive bays house the many kinds of storage devices a computer might contain.

Expansion Slots: An expansion slot is a slot located inside a computer mother board that allow additional peripherals to be connected to it.

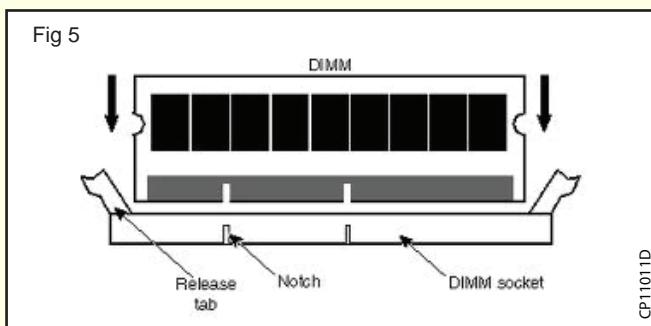
Fig 4



CP20N130714

Memory Slot

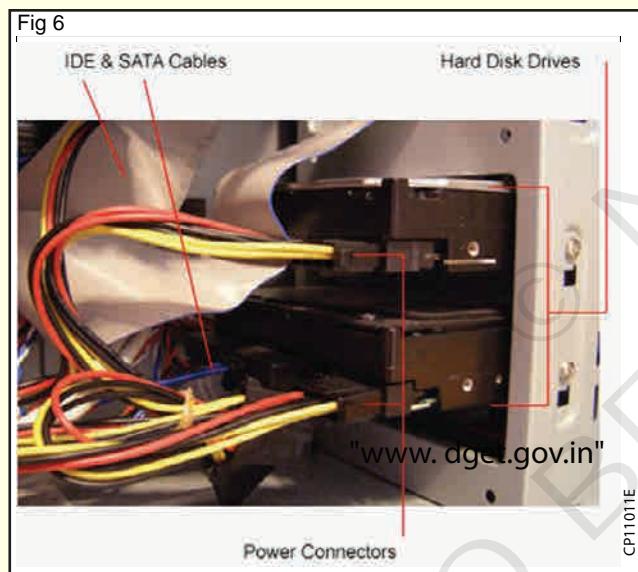
Memory Slot is used to insert a Random Access Memory(RAM) shown in Fig 5.



Storage Drivers

Storage drivers such as hard drives, optical drives and floppy drives all connect to the motherboard via cables and is mounted inside the computer.

IDE & SATA Cables: Fig 6 shows two hard disk drives that connect in different ways to the motherboard. One uses the older IDE cable connection while the other uses SATA(Serial Advanced Technology Attachment) cable which provides for faster hard drive access.



Power Port: Power is delivered to drives via cables that plug into the power port on the drives.

Peripheral cards slot

The peripheral cards are the spare expansion slots available on the mother board on which peripheral cards can be inserted.

The following are the peripheral cards

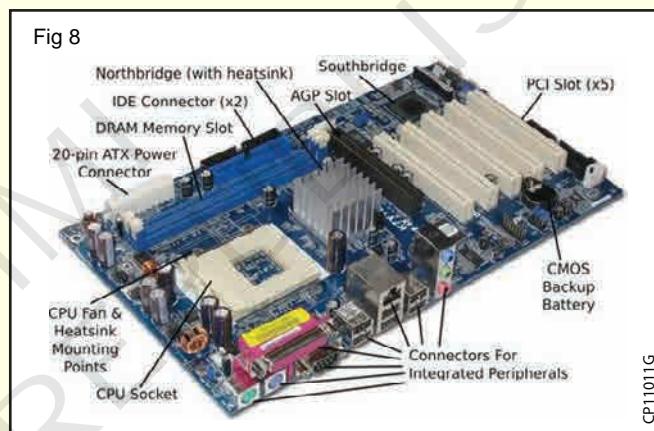
- Sound card
- Video card
- Modem
- Wireless network

Fig 7 shows the peripheral card, designed with a PCI (Peripheral component interconnect) connector.



Motherboard

The motherboard is a printed circuit that is the foundation of a computer and allows the CPU, RAM, and all other computer hardware components to function with each other as on Fig 8.



The motherboard is the primary component of the entire system. A mother board is a large board containing a number of tiny electronic circuits and other components are visible. All peripheral devices are connected to the motherboard. The components of the motherboard are:

- Keyboard / mouse port
- Parallel and Serial port
- Processor Socket
- AGP Slot
- PCI Slots
- ISA Slot
- CMOS Battery
- Data Card Connector
- Memory Slots
- Floppy Port
- Fan Header
- Main Power Connector

Floppy Port : The floppy drive connects to the computer via a 34-pin ribbon cable, which in turn connects to the motherboard. A floppy controller is one that is used to control the floppy drive.

RAM slots: Random-Access Memory (RAM) stores programs and data currently being used by the CPU.

RAM is measured in units called bytes. RAM has been packaged in many different ways

- SIMM-Single inline memory module -32 or 72 Pin
- DIMM- Dual Inline Memory module -168 pin.

In most of the PC's uses of the DIMM module

ROM BIOS Chip: This means Read Only Memory Basic Input-Output System.

The built-in software that determines what a computer can do without accessing programs from a disk. On PCs, the **BIOS** contains all the code required to control the keyboard, display screen, disk drives, serial communications, and a number of miscellaneous functions.

The BIOS is typically placed in a **ROM** chip that comes with the computer (it is often called a **ROM BIOS**). This ensures that the **BIOS** will always be available and will not be damaged by disk failures.

It also makes it possible for a computer to boot itself. Because RAM is faster than **ROM**, though, many computer manufacturers design systems so that the BIOS is copied from **ROM** to **RAM** each time the computer is booted. This is known as shadowing.

Many modern PCs have flash **BIOS**, which means that the **BIOS** have been recorded on a flash memory chip, which can be updated if necessary. The PC **BIOS** is fairly standardized, so all PCs are similar at this level (although there are different **BIOS** versions). Additional **DOS** functions are usually added through software modules.

This means you can upgrade to a newer version of DOS without changing the **BIOS**. PC **BIOS** that can handle Plug-and-Play (PnP) devices are known as PnPBIOS. These BIOS are always implemented with flash memory rather than ROM.

CMOS Battery

CMOS (Complementary Metal-Oxide-Semiconductor) is the term usually used to describe the small amount of memory on a computer motherboard that stores the **BIOS** settings.

Most **CMOS** batteries will last the lifetime of a motherboard (up to 10 years in most cases) but will sometimes need to be replaced. Incorrect or slow system date and time and loss of BIOS settings are major signs of a dead or dying CMOS battery.

ISA slot: (Industry Standard Architecture) It is the standard architecture of the Expansion bus. Motherboard may contain some slots to connect ISA compatible cards.

PCI slot : Intel introduced the Peripheral Component Interconnect bus protocol. The PCI bus is used to connect I/O devices to the main logic of the computer. **PCI** bus has replaced the ISA bus. PC motherboards have one PCI slot but generally more than one.

The **PCI** bus architecture is a processor-independent bus specification that allows peripherals to access system memory directly without using the CPU.

AGP slot: The Accelerated Graphics Port (AGP) is a high-speed point-to-point channel for attaching a video card to a computer's motherboard.

Power supply plug in

The Power supply, as its name implies, provides the necessary electrical power to make the PC (Personal Computer) operate. The power supply takes standard 110-V AC power and converts into +/-12-Volt, +/-5-Volt, and 3.3-Volt DC power. The power supply connector has 20-pins, and the connector can go in only one direction.

Hard Disk and Partitions

Partitioning is a process of dividing the Hard disk into several chunks, and uses any one of the portion or partition to install OS or use two or more partitions to install multiple OS..

But it can always have one partition, and use up the entire Hard disk space to install a single OS, but this will become data management nightmare for users of large Hard disks.

Now, because of the structure of the Master Boot Record (MBR), has only four partitions, and these four partitions are called **Primary Partitions**.

Extended Partition is not a usable partition by itself, but it's like a "container" and it is used to hold **Logical Drives!** That is this Extended Partition can be subdivided into multiple logical partitions.

In order to boot into a Partition, it must be designated as **bootable partition** or Active Partition. Active Partition is that partition which is flagged as bootable or which contains OS, this is generally a Primary Partition.

Types of Partitions:

- Master
- Partition
- Extended and
- Logical Extended

Master Boot Record (MBR): MBR is a small 512 bytes partition which is at the first physical sector of the hard disk. The location is denoted as CHS 0,0,1 meaning 0th Cylinder, 0th Head and 1st Sector.

MBR contains a small program known as bootstrap program which is responsible for booting into any OS. MBR also contains a table known as Partition Table.

This Partition Table is a table which lists the available Primary Partitions in the hard disk. Partition table considers whole Extended Partition as one Primary partition and lists it in the table!

So a Partition table can have two possible entries:-

- Up to 4 Primary Partitions.
- Up to 3 Primary Partitions and 1 Extended Partition. (Total not exceeding 4).

Partition Boot Sector (PBR): This is the logical first sector, that is sector at the start of a Primary Partition. This is also 512 byte area, which contains some programs to initialize or run OS files. All Primary Partitions have its own PBRs.

Extended Boot Sector (EBR): This is the logical first sector, that is the sector at the start of the Extended Partition. This EBR contains a Partition Table, which lists the available Logical Partitions inside Extended Partition. That is it contains the Starting addresses of each Logical Partitions.

Logical Extended Boot Sector (LEBR): This is the logical first sector residing at the start of each Logical Partition. This is similar to PBR for Primary Partitions.

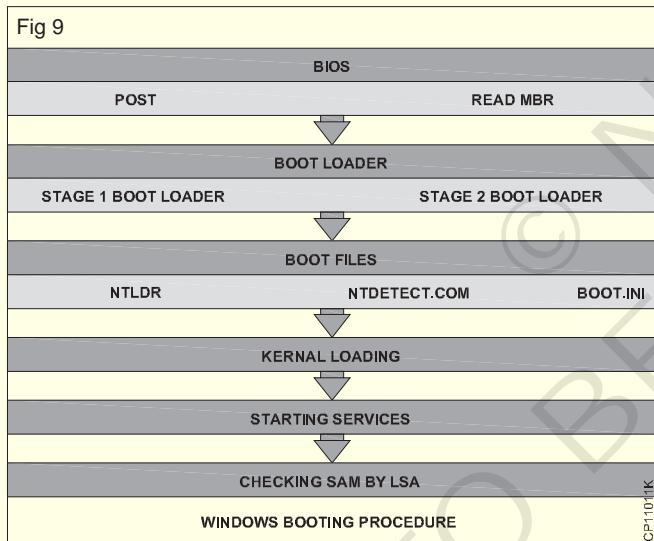
Booting

Booting is a process of loading the operating system (OS) and checking all the system software and hardware those are installed in the computer.

Booting procedure of Windows operating system

Functions of BIOS

The first process starts, when the computer switched on Basic Input Output System (BIOS) perform two functions, to conduct POST and read MBR (Fig 9).



POST - POST stands for Power OnSelf Test. POST checks all the hardware devices connected to a computer like RAM, hard disk etc and make sure that the system can run smoothly with those hardware devices. If the POST is a failure the system halts with a beep sound.

Now BIOS checks the **boot priority**. We can set the boot priority as CD drive, hard disk or floppy drive.

MBR - The next duty of BIOS is to read the MBR. MBR stands for Master Boot Record and it's the first sector on a hard disk. MBR contains the partition table and boot loader.

Functions of Boot loader

Now BIOS has passed the control to boot loader and boot loader is a small program which loads kernel to computers memory. Actually there are two stages of boot loaders, stage 1 boot loader and stage 2 boot loader.

The stage 1 boot loader is a link to the stage 2 boot loader. The stage 2 boot loader resides in the boot partition and it loads the kernel to memory.

Boot files and functions

There are three boot files in a Windows operating system and they are NTLDR, NTDETECT.COM and Boot.ini. The boot files are found in the active partition of hard disk and its normally C drive in a Windows machine.

NTLDR - NTLDR stands for NT Loader and it's the second stage boot loader. The path of NTLDR is C:\Windows\i386\NTLDR.

Boot.ini - Boot.ini contains the configuration files of NTLDR. When the operating system is loaded we cannot pass any arguments to kernel, so those arguments are passed through boot.ini. You can edit boot.ini by opening through notepad. The path of Boot.ini is C:\boot.ini.

NTDETECT.COM

This file detects hardware's and passes information to NTLDR. Using the collected information the NTLDR creates a hardware key and this key is used to detect hardware's.

A new hardware key is generated after each reboot of the operating system and that's why system asks to reboot after installation of a new hardware. The hardware keys created by NTLDR can be found in Windows registry at HKEY_LOCAL_MACHINE\HARDWARES.

View the BIOS settings and their modifications

Objective: At the end of this exercise you shall be able to

- list out the windows versions and features.
-

Windows versions and its features

Microsoft Windows has seen nine major versions since its first release in 1985. Over 29 years later, Windows looks very different but somehow familiar with elements that have survived the test of time, increases in computing power and - most recently - a shift from the keyboard and mouse to the touchscreen.

Here's a brief look at the history of Windows, from its birth at the hands of Bill Gates with Windows 1 to the latest arrival under new Microsoft chief executive SatyaNadella.

Windows 1

The first version of Windows

This is where it all started for Windows. The original Windows 1 was released in November 1985 and was Microsoft's first true attempt at a graphical user interface in 16-bit.

Development was spearheaded by Microsoft founder Bill Gates and ran on top of MS-DOS, which relied on command-line input.

It was notable because it relied heavily on use of a mouse before the mouse was a common computer input device. To help users become familiar with this odd input system, Microsoft included a game, Reversi (visible in the screenshot) that relied on mouse control, not the keyboard, to get people used to moving the mouse around and clicking onscreen elements.

Windows 2

Windows 2 with overlapping windows.

Two years after the release of Windows 1, Microsoft's Windows 2 replaced it in December 1987. The big innovation for Windows 2 was that windows could overlap each other, and it also introduced the ability to minimise or maximise windows instead of "iconising" or "zooming".

The control panel, where various system settings and configuration options were collected together in one place, was introduced in Windows 2 and survives to this day.

Microsoft Word and Excel also made their first appearances running on Windows 2.

Windows 3

Windows 3.0 got colourful.

The first Windows that required a hard drive launched in 1990. Windows 3 was the first version to see more widespread success and be considered a challenger to Apple's Macintosh and the Commodore Amiga graphical user interfaces, coming pre-installed on computers from

PC-compatible manufacturers including Zenith Data Systems.

Windows 3 introduced the ability to run MS-DOS programmes in windows, which brought multitasking to legacy programmes, and supported 256 colours bringing a more modern, colourful look to the interface.

More important - at least to the sum total of human time wasted - it introduced the card-moving timesink (and mouse use trainer) Solitaire.

Windows 3.1

Windows 3.1 with Minesweeper.

Windows 1 and 2 both had point release updates, but Windows 3.1 released in 1992 is notable because it introduced TrueType fonts making Windows a viable publishing platform for the first time.

Minesweeper also made its first appearance. Windows 3.1 required 1MB of RAM to run and allowed supported MS-DOS programs to be controlled with a mouse for the first time. Windows 3.1 was also the first Windows to be distributed on a CD-ROM, although once installed on a hard drive it only took up 10 to 15MB (a CD can typically store up to 700MB).

Windows 95

Windows 95: oh hello Start menu.

As the name implies, Windows 95 arrived in August 1995 and with it brought the first ever Start button and Start.

It also introduced the concept of "plug and play" - connect a peripheral and the operating system finds the appropriate drivers for it and makes it work. That was the idea; it didn't always work in practice.

Windows 95 also introduced a 32-bit environment, the task bar and focused on multitasking. MS-DOS still played an important role for Windows 95, which required it to run some programmes and elements.

Internet Explorer also made its debut on Windows 95, but was not installed by default requiring the Windows 95 Plus! pack. Later revisions of Windows 95 included IE by default, as Netscape Navigator and NCSA Mosaic were popular at the time.

Windows 98

Windows 98, the last great DOS-based Windows.

Released in June 1998, Windows 98 built on Windows 95 and brought with it IE 4, Outlook Express, Windows Address Book, Microsoft Chat and NetShow Player, which was replaced by Windows Media Player 6.2 in Windows 98 Second Edition in 1999.

Windows 98 introduced the back and forward navigation buttons and the address bar in Windows Explorer, among other things. One of the biggest changes was the introduction of the Windows Driver Model for computer components and accessories - one driver to support all future versions of Windows.

USB support was much improved in Windows 98 and led to its widespread adoption, including USB hubs and USB mice.

Windows ME

Windows ME was one to skip.

Considered a low point in the Windows series by many - at least, until they saw Windows Vista - Windows Millennium Edition was the last Windows to be based on MS-DOS, and the last in the Windows 9x line.

Released in September 2000, it was the consumer-aimed operating system twinned with Windows 2000 aimed at the enterprise market. It introduced some important concepts to consumers, including more automated system recovery tools.

IE 5.5, Windows Media Player 7 and Windows Movie Maker all made their appearance for the first time. Autocomplete also appeared in Windows Explorer, but the operating system was notorious for being buggy, failing to install properly and being generally poor.

Windows 2000

Windows 2000 was ME's enterprise twin.

The enterprise twin of ME, Windows 2000 was released in February 2000 and was based on Microsoft's business-orientated system Windows NT and later became the basis for Windows XP.

Microsoft's automatic updating played an important role in Windows 2000 and became the first Windows to support hibernation.

Windows XP

Windows XP still survives to this day.

Arguably one of the best Windows versions, Windows XP was released in October 2001 and brought Microsoft's enterprise line and consumer line of operating systems under one roof.

Advertisement

It was based on Windows NT like Windows 2000, but brought the consumer-friendly elements from Windows ME. The Start menu and task bar got a visual overhaul, bringing the familiar green Start button, blue task bar and vista wallpaper, along with various shadow and other visual effects.

ClearType, which was designed to make text easier to read on LCD screens, was introduced, as were built-in CD burning, autoplay from CDs and other media, plus various automated update and recovery tools, that unlike Windows ME actually worked.

Windows XP was the longest running Microsoft operating system, seeing three major updates and support up until April 2014 - 13 years from its original release date. Windows XP was still used on an estimated 430m PCs when it was discontinued.

Its biggest problem was security: though it had a firewall built in, it was turned off by default. Windows XP's huge popularity turned out to be a boon for hackers and criminals, who exploited its flaws, especially in Internet Explorer, mercilessly - leading Bill Gates to initiate a "Trustworthy Computing" initiative and the subsequent issuance of Service Pack updates that hardened XP against attack substantially.

Windows Vista

Windows Vista, arguably worse than Windows ME.

Windows XP stayed the course for close to six years before being replaced by Windows Vista in January 2007. Vista updated the look and feel of Windows with more focus on transparent elements, search and security. Its development, under the codename "Longhorn", was troubled, with ambitious elements abandoned in order to get it into production.

Advertisement

It was buggy, burdened the user with hundreds of requests for app permissions under "User Account Control" - the outcome of the Trustworthy Computing initiative which now meant that users had to approve or disapprove attempts by programs to make various changes.

The problem with UAC was that it led to complacency, with people clicking "yes" to almost anything - taking security back to the pre-UAC state. It also ran slowly on older computers despite them being deemed as "Vista Ready" - a labelling that saw it sued because not all versions of Vista could run on PCs with that label.

PC gamers saw a boost from Vista's inclusion of Microsoft's DirectX 10 technology.

Windows Media Player 11 and IE 7 debuted, along with Windows Defender an anti-spyware programme. Vista also included speech recognition, Windows DVD Maker and Photo Gallery, as well as being the first Windows to be distributed on DVD. Later a version of Windows Vista without Windows Media Player was created in response to anti-trust investigations.

Windows 7

Windows 7 was everything Windows Vista should have been.

Considered by many as what Windows Vista should have been, Windows 7 was first released in October 2009. It was intended to fix all the problems and criticism faced by Vista, with slight tweaks to its appearance and a concentration on user-friendly features and less "dialogue box overload".

It was faster, more stable and easier to use, becoming the operating system most users and business would upgrade to from Windows XP, forgoing Vista entirely.

Handwriting recognition debuted in 7, as did the ability to "snap" windows to the tops or sides of the screen, allowing faster more automatic window resizing.

Windows 7 saw Microsoft hit in Europe with antitrust investigations over the pre-installing of IE, which led to a browser ballot screen being shown to new users allowing them to choose, which browser to install on first boot.

Windows 8

Windows 8 focused more on touch than a keyboard and mouse.

Released in October 2012, Windows 8 was Microsoft's most radical overhaul of the Windows interface, ditching the Start button and Start menu in favour of a more touch-friendly Start screen.

Advertisement

The new tiled interface saw programme icons and live tiles, which displayed at-a-glance information normally associated with "widgets", replace the lists of programmes and icons. A desktop was still included, which resembled Windows 7.

Windows 8 was faster than previous versions of Windows and included support for the new, much faster USB 3.0 devices.

The Windows Store, which offers universal Windows apps that run in a full-screen mode only, was introduced. Programs could still be installed from third-parties like other iterations of Windows, but they could only access the traditional desktop interface of Windows.

The radical overhaul was not welcomed by many. Microsoft attempted to tread a fine line between touchscreen support and desktop users, but ultimately desktop users wanting to control Windows with a traditional mouse and keyboard and not a touchscreen felt Windows 8 was a step back.

There were also too few touchscreens in use, or on offer, to make its touch-oriented interface useful or even necessary - despite the parallel rise of tablets such as the iPad, and smartphones, which had begun outselling PCs by the end of 2010.

Windows RT, which runs on ARM-based processors traditionally found in smartphones and non-PC tablets, was introduced at the same time as Windows 8 with the Microsoft Surface tablet.

It looked and felt like Windows 8, but could not run traditional Windows applications, instead solely relying on the Windows Store for third-party apps.

Windows 8.1

Windows 8.1 and the great reappearance of the Start button: A free point release to Windows 8 introduced in October 2013, Windows 8.1 marked a shift towards yearly software updates from Microsoft and included the first step in Microsoft's U-turn around its new visual interface.

Windows 8.1 re-introduced the Start button, which brought up the Start screen from the desktop view of Windows 8.1. Users could also choose to boot directly into the desktop of Windows 8.1, which was more suitable for those using a desktop computer with a mouse and keyboard than the touch-focused Start screen.

Windows 10

With Windows 10, Microsoft is trying to keep some of the touch and tablet features it created for Windows 8, combine them with the familiar Start menu and desktop, and run it all on top of an improved operating system with more security, a new browser, the Cortana assistant, its own version of Office for on-the-go editing and plenty of new features intended to make life simpler.

Of course, that also means it's very different to use, whether you come from Windows 7, Windows 8 or Windows XP. You have to look in a new place even to turn your PC off.

On top of that, Windows 10 is more than just a PC operating system; it's also what will run on Windows phones - and on small tablets as well, because a 6-inch phone and a 7-inch tablet aren't such very different devices.

Microsoft is expecting people to put Windows 10 on a billion devices (which ought to encourage more app developers to at least take a look at building their apps for Windows phones and tablets, as well as for Xbox One and HoloLens).

The Start menu evolves

The full-screen Start screen of Windows 8 is back to being a Start menu in Windows 10 that tries to combine the best of both options. A scrolling Start menu that's restricted to a single column, with jump lists and flyout menus for extra options, divided into frequently used and recently installed programs, with the option to switch to a scrolling view of all the applications, sorted alphabetically.

But also get an extra pane, where can pin Windows 8-style tiles, complete with 'rotating 3D cube' animations of live tiles. drag the Start menu to be a larger size or even set it to be full screen.

Desktop Background

Another component of the Desktop is the Background. This is simply an image that appears at the back of the screen. Most computers come with a pre-selected background, but change it to any image.

Fig 1



To change the background, follow these steps:

- 1 Right-click on the background and choose "Personalize"
- 2 From the Personalization window, choose from a series of pre-selected pictures or browse for your own.

After choosing a picture, the Background will change automatically.

Start Menu

If looking for a specific application, open the Start Menu and click "All Applications". This will open an alphabetical list of all the applications installed on computer.

Fig 2



Fig 4



Fig 3

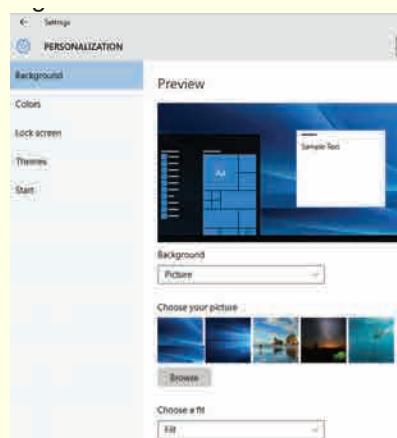
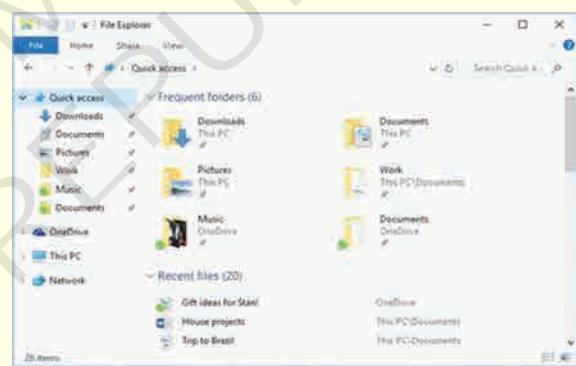


Fig 5



Fig 6



File Explorer

If you are looking for a specific document, another alternative is to use the File Explorer by clicking on the Folder icon on the Taskbar.

In the File Explorer window, browse all the folders and documents.

Virtual Desktops

One of the new features of Windows 10 is the addition of Virtual Desktops. This allows you to have multiple desktop screens where to keep open windows organized.

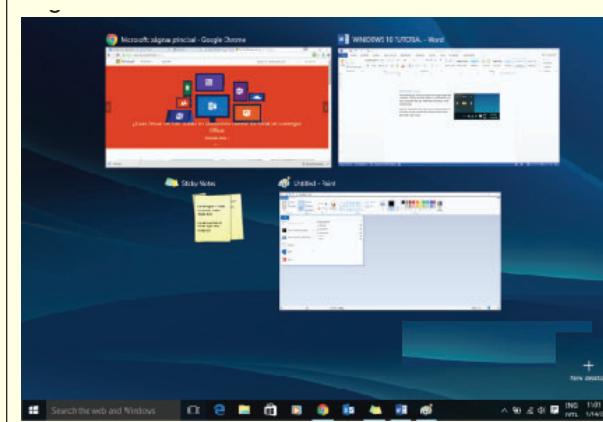
To add a virtual desktop, follow these steps:

- 1 Click Task View on the Taskbar
- 2 Click the "New desktop" option on the lower-right corner. You can access or delete the new Desktop by clicking Task View again.

Cortana helps as with search and control

Cortana, the Windows Phone assistant, shows up in Windows 10 as a search pane on the taskbar, which can also trigger by saying 'Hey Cortana' - and when start searching the Start menu.

Fig 7



That gets the apps have installed, documents access to, apps could install from the Store, search results from the web and a range of other information - including from apps and services that integrate with Cortana.

Fig 8



Copa 13708

Activating Cortana

If Cortana isn't active, can turn it on by typing "Cortana" in the Taskbar search to access the Cortana settings, or just clicking the "Gear" icon on the left-side of the menu.

Fig 9



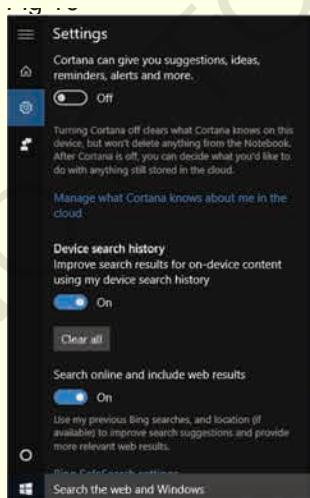
Copa 13709

After activating Cortana, it will start gathering information about to personalize the experience.

Task switcher

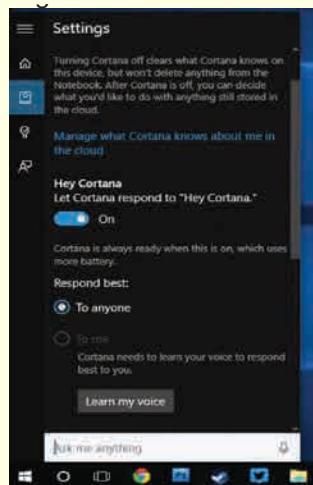
Most Windows users don't know the Alt-Tab keyboard combination to see and switch between all running apps, so as well as having a redesigned task switcher with bigger thumbnails, Windows 10 also puts a task view icon in the taskbar to help them find it.

Fig 10



Search the web and Windows

Fig 11



Copa 13701

Taskbar: The Windows 10 taskbar sits at the bottom of the screen giving the user access to the Start Menu, as well as the icons of frequently used applications. On the right-side, the Taskbar features the Notification Area which informs the user of different things like the state of the Internet connection or the charge of the laptop battery.

The icons in the middle of the Taskbar are "pinned" applications, which is a way to have a quick access to applications you use frequently. "Pinned" applications will stay in the Taskbar until you "unpin" them.

Fig 12



Copa 1137012

Step 1: Search for the application you want to pin in the Start Menu.

Step 2: Right-click on the application.

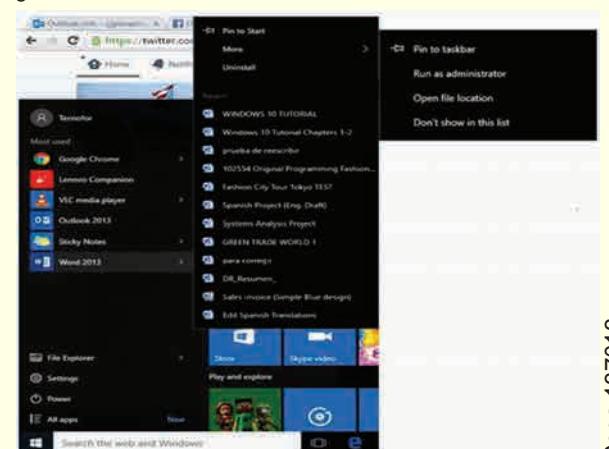
Step 3: Select "More" option at the top of the menu.

Step 4: Select the "Pin to taskbar" option.

Unpin an Application from the Taskbar

To "unpin" it, just right-click the icon in the Taskbar and select "Unpin from taskbar". You can "pin" it back again any time you want.

Fig 13



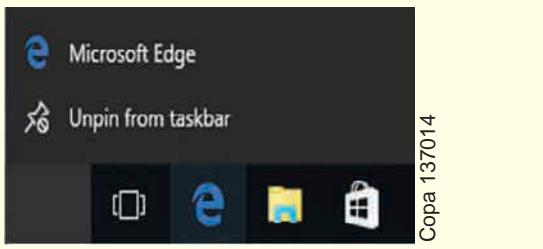
Copa 137013

Notification Area

The Notification Area is located at the right side of the Taskbar. It shows different types of notifications from your computer like your Internet connection, or the volume level.

At first, the Notification Area shows a limited amount of icons. But you can click the upward arrow on its left-side to see other icons as well.

Fig 14

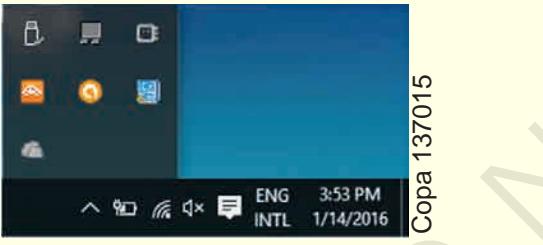


Copa 137014

Snap Assist

Because all the apps and programs run in windows on the desktop, instead of modern apps from the Store being in their own space, and can no longer drag across the left edge of the screen to bring another app on screen and get a split view. Instead, drag windows into the corners of the screen to get the familiar Snap view.

Fig 15



Copa 137015

Now use all four corners of the screen if want each window to take up a quarter of the screen instead of half, and the space that isn't filled by the window you just dragged shows thumbnails of your other windows to make it easier to snap the next one into place.

Action Center

If we used Windows Phone 8.1 (or Android and/or iOS), we used to a notification centre can drag down from the top of the screen.

Windows 10 puts that on the right of the screen, where the charms bar was in Windows 8, with notifications from various apps at the top and the choice of various settings buttons at the bottom for quick access.

The command prompt

Those of us that use the command prompt have been stuck with pretty much the same experience since the 1990s, but in Windows 10 can finally resize the command prompt window and use familiar keyboard shortcuts to copy and paste at the command prompt.

It's far from ground-breaking but it's a very welcome improvement after years of frustration.

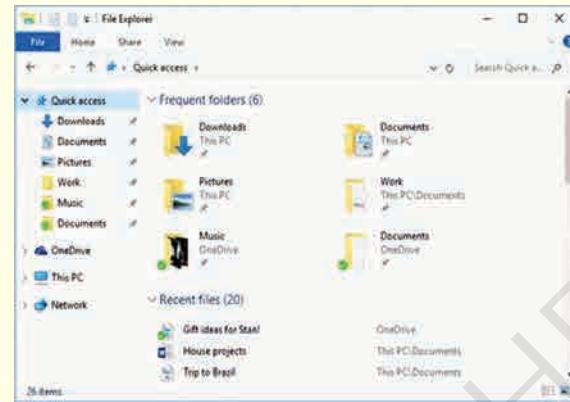
File explorer

File Explorer is the file management application used by Windows operating systems to browse folders and files. It

provides a graphical interface for the user to navigate and access the files stored in the computer.

The main way to access the File Explorer is by clicking the folder icon in the Taskbar. After clicking the icon, the

Fig 16



Copa 137015

File Explorer window will open.

The initial File Explorer window is comprised of the following sections:

- 1 The **File Explorer ribbon**, which resembles the ribbon featured in Microsoft Office. The ribbon contains buttons for common tasks to perform with your files and folders.
- 2 The **Navigation Pane** gives you access to your libraries of documents and pictures, as well as your storage devices. It also features frequently used folders and network devices.
- 3 The **Frequent folders section** on the right features the folders you've worked with recently to allow for quick access to them.
- 4 The **Recent files section** in the lower part of the window features files and documents that you've opened recently.

The new Edge browser

To catch up with fast-moving browsers like Chrome and Firefox, Microsoft took its browser back to basics, ripping out years of code that didn't fit with web standards and making a lean, fast browser.

It's a work in progress - it won't get support for things like ad-blocking extensions until a while after Windows 10 launches - but can do plenty of neat things here. For example, and can scribble notes on a web page to send to a friend and Edge has Cortana built in to pull useful information out of web pages, like the phone number of a restaurant, or the opening hours.

Sites like Medium that didn't work properly with IE should look better and have more features in Edge.

Multiple desktops

User need to arrange a lot of windows and don't have multiple monitors, user can put them on multiple virtual desktops. And can use Alt-Tab to move between apps as usual and then Windows-Ctrl and the left and right arrow keys to move between desktops.

Schedule restarts

No more having Windows announce that you have fifteen minutes to get everything done before it restarts to apply an update. Instead of leaving Windows 10 to decide when to do that, if there's an update that will need a restart and can have Windows ask when you want to schedule that for.

User can only do that once the update has been downloaded. If user want to have certain times off-limits for restarts, they will need the features in Windows Update for Business (for Windows 10 Pro and Enterprise) which lets block restarts so they don't happen in working hours, or on certain dates.

Universal apps - including Office

Windows 10 gets a new Windows Store, where download desktop programs as well as modern Windows apps. Many of those apps will be universal apps that are the same code on a PC, a Windows phone, an Xbox One and even

on HoloLens, with the interface changing to suit the different screen sizes. The Office for Windows apps like Word and Excel are universal apps, as are the Outlook Mail and Calendar apps.

Settings and control panel

The Windows 8 Settings app has taken over many more of the settings that used to be in Control Panel, and it has a Control Panel-style interface with icons to navigate with. But the old Control Panel interface is still there, for settings that aren't in the new Settings app (or if you're just used to finding things there).

Windows 10 - Keyboard Shortcuts ITES

Like most Windows applications, there are several keyboard shortcuts you can use to make it easier or faster for some to perform certain tasks.

Most of the new Windows shortcuts use the Windows key () combined with other keys to perform several actions. The following are some of the most common or useful shortcuts used in Windows 10.

Keyboard Shortcuts for Navigating Windows 10

Press This	To Do This
Windows Logo	Toggle the Start menu
Windows Logo+A	Open the Notifications pane
Windows Logo+B	Activate the notification area's Show Hidden Icons arrow (press Enter to display the hidden icons)
Windows Logo+C	Open Cortana for voice commands
Windows Logo+D	Minimize all open windows to display the desktop
Windows Logo+E	Run File Explorer
Windows Logo+F	Display the Start menu and activate the Search box
Windows Logo+H	Display the Share pane
Windows Logo+I	Run the Settings app
Windows Logo+K	Display the Devices pane
Windows Logo+L	Lock your computer
Windows Logo+M	Minimize all windows
Windows Logo+O	Turn the tablet orientation lock on and off
Windows Logo+P	Display the Project pane to configure a second display
Windows Logo+Q	Open Cortana for voice commands
Windows Logo+R	Open the Run dialog box
Windows Logo+S	Open Cortana for keyboard commands
Windows Logo+T	Activate the taskbar icons (use the arrow keys to navigate the icons)
Windows Logo+U	Open the Ease of Access Center
Windows Logo+W	Activate the Search box
Windows Logo+X	Display a menu of Windows tools and utilities
Windows Logo+Z	Display an app's commands (although this works in only some Modern apps)
Windows Logo+=	Open Magnifier and zoom in
Windows Logo+-	Zoom out (if already zoomed in using Magnifier)

Windows Logo+,	Temporarily display the desktop
Windows Logo+Enter	Open Narrator
Windows Logo+Left	Snap the current app to the left side of the screen
Windows Logo+Right	Snap the current app to the right side of the screen
Windows Logo+Up	Restore a minimized app; maximize a restored app
Windows Logo+Down	Restore a maximized app; minimize a restored app
Windows Logo+PgUp	Move the current app to the left monitor
Windows Logo+PgDn	Move the current app to the right monitor
Windows Logo+PrtSc	Capture the current screen and save it to the Pictures folder
Windows Logo+Ctrl+D	Create a virtual desktop
Windows Logo+Ctrl+Right	Switch to the next virtual desktop
Windows Logo+Ctrl+Left	Switch to the previous virtual desktop
Windows Logo+Ctrl+F4	Close the current virtual desktop
Windows Logo+Tab	Open Task View, which displays thumbnails for each running app as well as the available virtual desktops

Install Windows operating system

Objectives : At the end of this lesson you shall be able to

- describe the functions of BIOS and CMOS
- describe the method of viewing and changing BIOS settings
- describe the meaning of partitioning and formatting
- describe the process of installing Windows operating system.

The Basic Input/Output System (BIOS), also known as System BIOS, ROM BIOS or PC BIOS is a generally accepted standard defining a firmware interface.

The fundamental purpose of the BIOS is to initialize and test the system hardware components and load an operating system from a mass memory device. The BIOS is special software that interfaces the major hardware components of the computer with the operating system. It is usually stored on a Flash memory chip on the motherboard, but sometimes the chip is another type of ROM. The BIOS is a firmware (software instructions permanently recorded on a chip located on your motherboard). (Refer Fig.1).

Fig 1



Copa 181301

CMOS Setup

The first thing the BIOS will do is check the information stored in a tiny (64 bytes) amount of RAM located on a complementary metal oxide semiconductor (CMOS) chip. The CMOS Setup provides detailed information particular to your system and can be altered as your system changes. The BIOS uses this information to modify or supplement its default programming as needed.

Configuring BIOS

The BIOS checks the CMOS Setup for custom settings. To change the CMOS settings we need to enter the CMOS setup. To enter the CMOS Setup, a certain key or combination of keys must be pressed during the initial startup sequence. Most systems use "Esc," "Del," "F1," "F2," "Ctrl-Esc" or "Ctrl-Alt-Esc" to enter setup. There is usually a line of text at the bottom of the display that tells "Press ___ to Enter Setup."

The BIOS setup shows a set of text screens with a number of options. Some of these are standard, while others vary according to the BIOS manufacturer. Common options include:

- System Time/Date - Set the system time and date
- Boot Sequence - The order that BIOS will try to load the operating system
- Plug and Play - A standard for auto-detecting connected devices; should be set to "Yes" if your computer and operating system both support it
- Mouse/Keyboard - "Enable Num Lock," "Enable the Keyboard," "Auto-Detect Mouse"...
- Drive Configuration - Configure hard drives, CD-ROM and floppy drives
- Memory - Direct the BIOS to shadow to a specific memory address
- Security - Set a password for accessing the computer
- Power Management - Select whether to use power management, as well as set the amount of time for "standby" and "suspend"
- Exit - Save your changes, discard your changes or restore default settings.

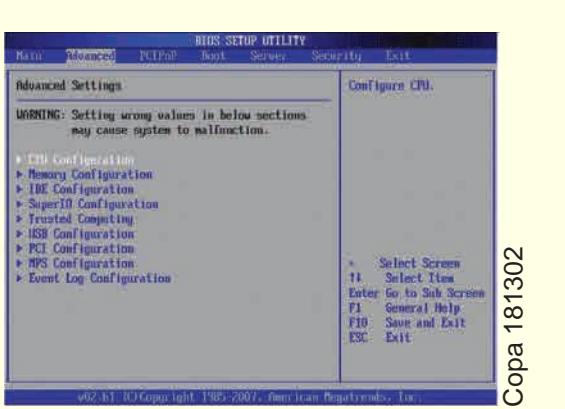
Functions of BIOS

The BIOS software has a number of different roles, but its most important role is to load the operating system. The BIOS checks and initializes the PC hardware each time the system powers up or restarts before handing over control to the operating system. Some of the other common tasks that the BIOS performs include:

- A power-on self-test (POST) for all of the different hardware components in the system to make sure everything is working properly
- Activating other BIOS chips on different cards installed in the computer - For example the graphics cards often have their own BIOS chips.
- Providing a set of low-level routines that the operating system uses to interface to different hardware devices. They manage things like the keyboard, the screen, and the ports, especially when the computer is booting.
- Managing a collection of settings for the hard disks, clock, etc.

The BIOS uses CMOS technology to save any changes made to the computer's settings. With this technology, a small lithium or Ni-Cad battery can supply enough power to keep the data for years. Major BIOS manufacturers include American Megatrends Inc. (AMI), Phoenix Technologies, Winbond etc. A typical BIOS screenshot is shown in fig. given below.(Refer Fig.2)

Fig 2



Copa 181302

Installing the Windows operating System

A hard disk needs to be partitioned (though not mandatory) and formatted before you can store data on it.

Partitioning

A partition, sometimes also called a volume, is an area on a hard disk that can be formatted with a file system and identified with a letter of the alphabet. For example, drive C on most Windows computers is a partition. the first three partitions you create are primary partitions. These can be used to start an operating system. If you want to create more than three partitions, the fourth partition is created as an extended partition.

An extended partition is a container that can hold one or more logical drives. Logical drives function like primary partitions except that they cannot be used to start an operating system.

Many computers are partitioned as a single partition that equals the size of the hard disk. Partitioning a hard disk into several smaller partitions is not required, but it can be useful for organizing data on your hard disk.

Creating more than one partition has the following advantages:

- Separation of the operating system (OS) and program files from user files.
- Having a separate area for operating system virtual memory swapping/paging.
- Keeping frequently used programs and data near each other.

- Use of multi-boot setups, which allow users to have more than one operating system on a single computer. For example, one could install Linux and Microsoft Windows or other operating systems on different partitions of the same HDD and have a choice of booting into any operating system at power-up.
- Protecting or isolating files, to make it easier to recover a corrupted file system or operating system installation. If one partition is corrupted, other file systems may not be affected.
- Raising overall computer performance on systems where smaller file systems are more efficient.
- Partitioning for significantly less than the full size available can reduce the time for diagnostic tools such as checkdisk to run.

Formatting

Disk formatting is the process of preparing a data storage device such as a hard disk drive, solid-state drive or USB flash drive for initial use. It is the act of creating a file system on a volume, so that the operating system can store and retrieve data on that volume.

Formatting of a disk is of two categories:

- 1 Low-level formatting (i.e., closest to the hardware) marks the surfaces of the disks with markers indicating the start of a recording block. It also provides information about block checks done for future use by the disk controller to read or write data. This is intended to be the permanent foundation of the disk, and is often completed at the factory. A hard disk needs to be partitioned and formatted before you can store data on it
- 2 High-level formatting creates the file system format within a disk partition or a logical volume. This formatting includes the data structures used by the OS to identify the logical drive or partition's contents. This may occur during operating system installation, or when adding a new disk.

Installing the Windows operating System

The three basic types of windows installation procedures are as follows:

- Install on a brand new disk or computer system
- Erase the disk, format it, and install.
- Install into a new directory for dual-booting

For the first two methods, it must be ensured that the computer can boot from a DVD or any other removable drive. To do this the drive boot order needs to be changed in the BIOS. The latest Windows DVDs are bootable and run the Setup program automatically. Then the installation can be done by following the procedure step by step as indicated on the subsequent screens as in trade practicals.

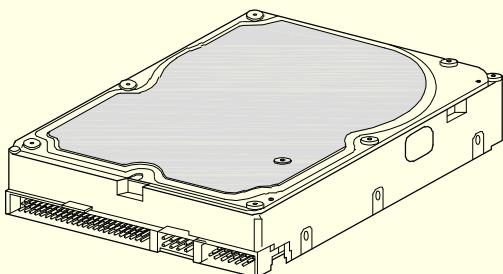
Format hard disk and create partition

Objectives : At the end of this lesson you shall be able to

- describe the functions of BIOS and CMOS
- describe the method of viewing and changing BIOS settings
- describe the meaning of partitioning and formatting
- describe the process of installing Windows operating system.

Format a hard drive

Fig 1



CP20N141611

There are many reasons why it is required to format a hard drive, such as to install Windows fresh, to get rid of a virus or malware or simply because a pc is refreshed or cleaned up on selling.

The process can be different depending on whether it's an only hard drive and whether there is a spare PC or not.

It cannot be formatted, the hard drive on which Windows is running. In order to do this, it is in need to boot the PC from a Windows installation disc, a USB flash drive or another bootable disc.

Format a disk?

Formatting is the process of deleting all the data on the hard drive, but beware of 'Quick Format' which leaves all data in place and makes the drive appear to be empty. A quick format is ok if there is a brand new hard drive, or need to reinstall Windows, but not if its disposing of the disk or giving it to someone else.

A word of warning: make sure to have successfully backed up any photos, videos, music and other documents from the drive before formatting it. Although deleted files can be recovered in some situations, prevention is always better than cure.

Format hard drive partitions

It's important to understand about partitions before getting started. A hard drive can be divided up into smaller sections, called partitions. It's possible to format one partition while leaving the others untouched.

If it is required to format the entire hard drive and use the entire capacity in one block, delete the partition information.

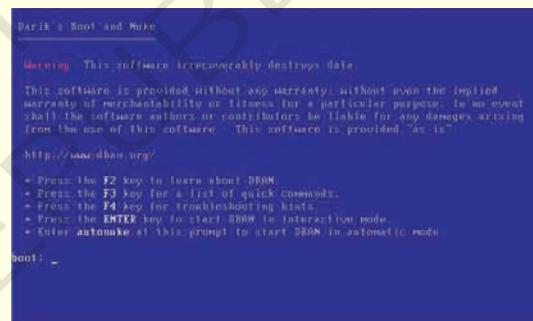
Format a hard drive from the BIOS?

Many people ask how to format a hard disk from BIOS. The short answer is no.

If it is required to format a disk and you can't do it from within Windows, create a bootable CD, DVD or USB flash drive and run a free third-party formatting tool.

One option is Darik's Boot and Nuke (DBAN), which is free for personal use. This program will totally erase and format your hard disk, allowing for a clean install of a new OS, but the process cannot be undone.

Fig 2



Copa 110102

DBAN is supposedly only able to create a bootable CD/DVD-R, but if don't have any blank discs or a burner, there is a workaround available in the form of a separate third-party program.

Universal USB Installer will quickly and easily convert the DBAN ISO image downloaded to run from a bootable USB. Simply insert a blank USB flash drive, run the Universal USB Installer setup program, and follow the prompts.

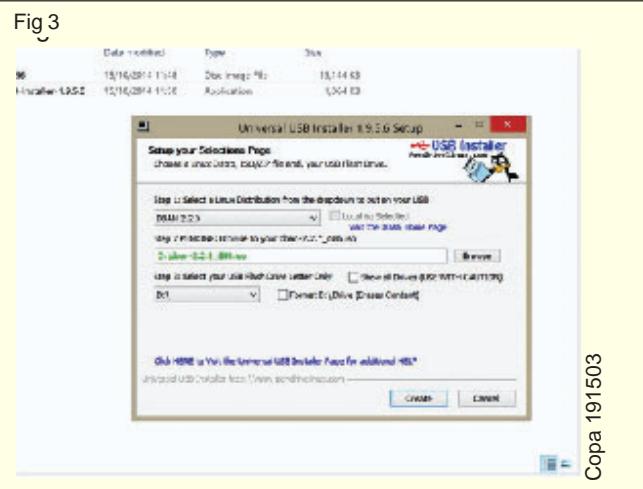
It will ask to scroll through and pick the Linux Distribution want to install to USB (in this case, the latest version of DBAN), followed by its location on the computer and the letter of the USB drive like to install it to.

Once that information is complete, click create.

To boot from this USB drive rather than usual boot device (in most cases, this would be the hard drive), it have to be changed some settings in the BIOS.

In the BIOS, navigate to the boot order settings, and change the primary boot device to the USB drive (it shouldn't need to be plugged in to make this selection). After saved the settings and exited the BIOS, insert the bootable USB, restart the computer.

Fig 3



Copa 191503

Fig 4



Copa 191504

The PC should automatically boot the DBAN software, which will guide through the process of erasing the hard drive, with options for different levels of data-wiping.

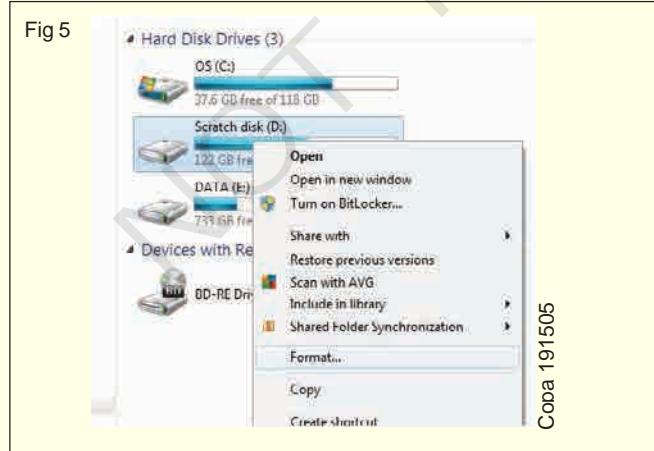
It will treat the USB as another drive so to avoid inadvertently wiping that as well, remove it after booted into DBAN.

Quick format a hard drive?

Yes, but don't use this method if want the data to be permanently erased. A quick format doesn't delete the data but instead erases only the pointers to the files.

Windows Vista, 7, 8 and 10 have a built-in Disk Management tool (see below), but the fastest way to format a hard drive is to click the Start button, then Computer and right-click on the hard drive to wipe. It can't be formatted the drive on which Windows is installed for obvious reasons.

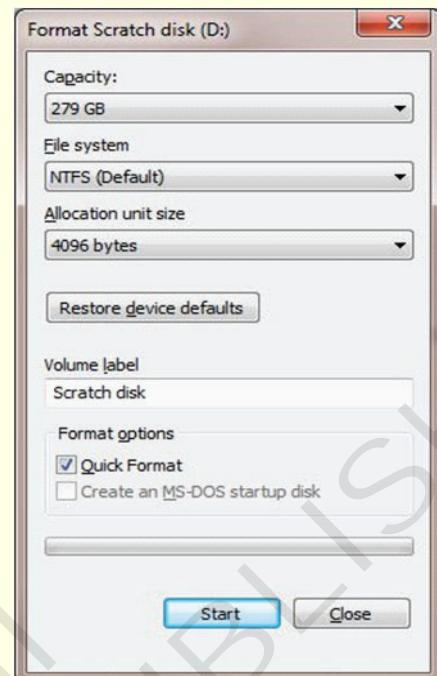
Fig 5



Copa 191505

By default Quick Format is checked, and choose the file system and allocation unit size as well as changing the volume label (the drive's name). Typically, leave all settings unchanged and click the Start button. In under a minute the hard drive will be formatted.

Fig 6



Copa 191506

Choose NTFS as the file system if it isn't already selected for Windows Vista, 7, 8 or 10 and ensure the Allocation Unit Size is set to 'Default Allocation Size'.

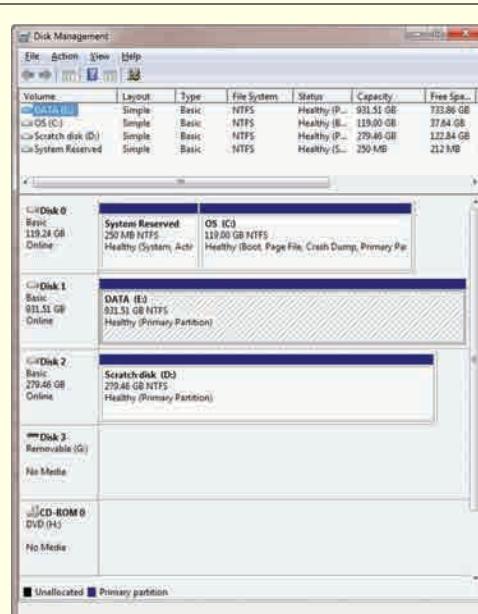
Using the Disk Management tool

Type diskmgmt.msc or Disk Management into the search box in Vista, 7, 8 or 10 and then click on only result that appears in the menu above, with the same name.

This is the easiest way to launch Disk Management, but also find it in the Control Panel if search for 'disk' and select the 'Create and format hard disk partitions'.

Disk Management isn't as powerful as a standalone partition management tool, but it is still capable of formatting data.

Fig 7

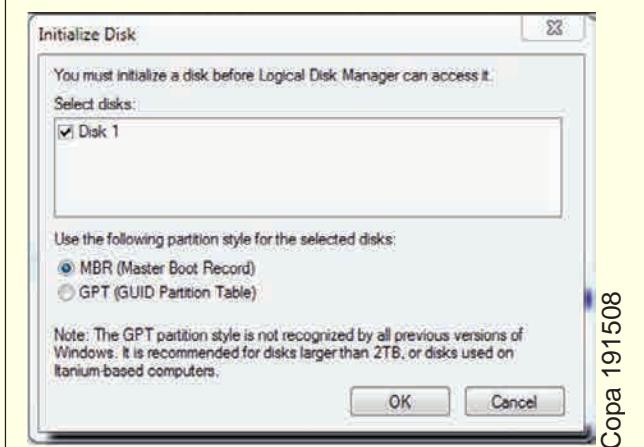


Copa 191507

If it is to install a new (additional) hard drive in a PC, it might be a thing to wonder why it doesn't appear in Windows Explorer. The reason is because it needs to be initialised and formatted -which can be done in Disk Management.

When the tool loads, it will analyse all of the computer's drives and will prompt to initialise a new disk that it finds.

Fig 8



Copa 191508

If the disk is larger than 2TB, opt for GPT (GUID Partition Table). This setting also lets to create more than four partitions.

If don't see a prompt, look in the list of drives and see one that says 'Not Initialized'. Right-click on it and choose Initialize Disk.

Once that's done, right-click in the hatched Unallocated space and choose New Simple Volume...

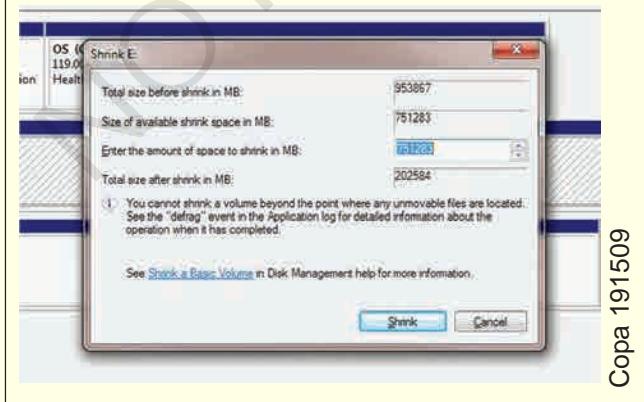
Follow the instructions, choosing how big to want the partition to be (in MB - 1024MB = 1GB), and which drive letter to assign (one will be chosen, but can opt to change it if desired).

If select a size for the partition that's smaller than the total capacity of the drive, say 500B on a 1TB drive, end up with some unallocated space on the drive which can format by repeating the process just completed.

change partition size

Disk Management can be used to expand or shrink a partition. Simply right-click on one and choose the appropriate option from the menu that appears. If shrinking, the partition will be checked to find out how much empty space it contains.

Fig 9



Copa 191509

It's a little confusing as the numbers are displayed in MB rather than GB, but it can be adjusted the amount of space to shrink and the 'Total size after shrink' will be updated, also can't shrink a partition beyond the point where files are located - it may be able to free up space by defragmenting the drive first.

Conversely it can only expand a partition if there is unused space on the drive. If not, the option will be greyed out.

Resetting Windows

If PC is having problems and the user don't wish to lose the personal files by wiping the hard drive, then it might want to refresh or reset the PC which can be done in Windows 8 and 10.

If user looking to keep your personal files and settings, but want to have a fresh Windows install, it'll want to refresh the PC.

Do note that a refresh will remove all programs and apps installed on the machine, but will keep the Windows-default programs intact.

A reset reinstalls Windows and deletes all files, settings and programs, it is suggested performing this if have previously backed up all the files and don't mind transferring user's personal files.

Windows 10 has a slightly different approach and might confuse those coming from Windows 8. Microsoft removed the refresh option and has instead combined the refresh and reset options into one setting.

To find the option, open the Start Menu, click on Settings > Update & security > Recovery > Get started (under the Reset this PC option).

Upon selecting the option, it will present with three separate options:

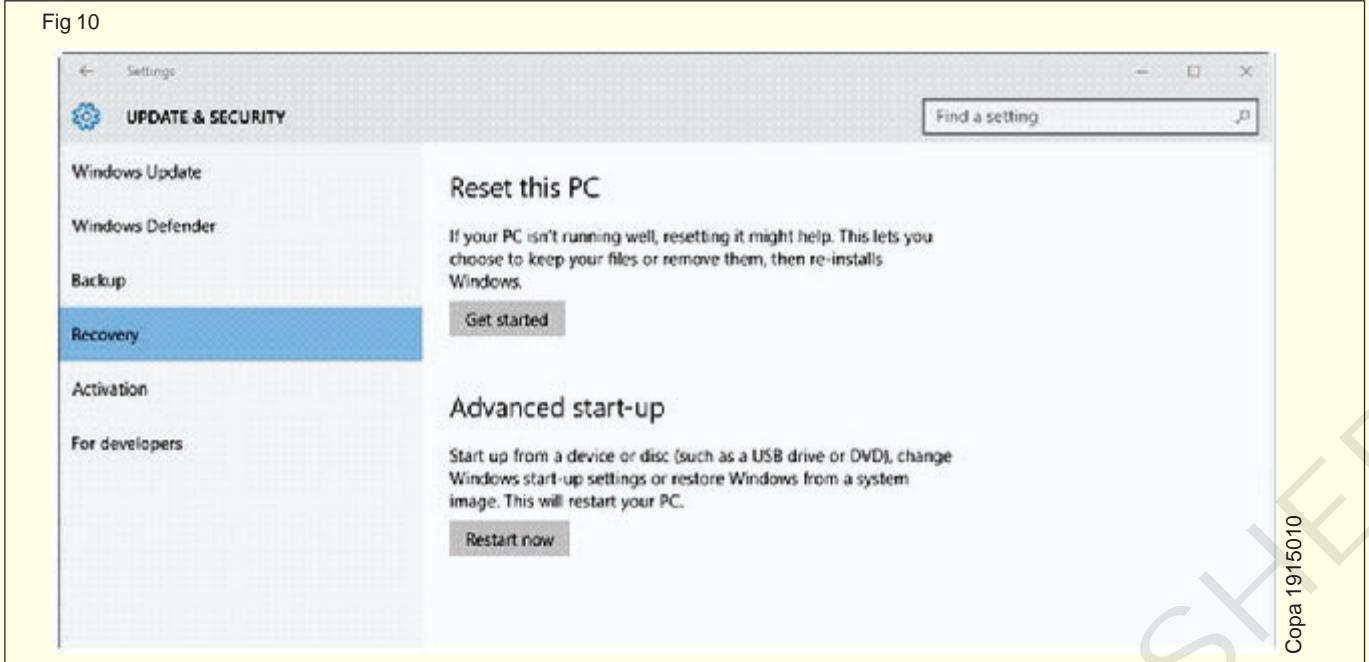
- Keep my files
- Remove everything
- Restore factory settings (not available on all PCs)

'Keep my files' saves personal files but deletes user settings, drivers and programs, whilst also reinstalling Windows 10.

'Remove everything' also gets rid of personal files and 'Restore factory settings' does the same actions as 'Remove everything', but also resets the PC to the version of Windows which came with the machine.

This option will only be available to those who bought a pre-built PC or laptop which came with Windows already pre-installed.

Fig 10



Identify and rectify common hardware and software issues during OS installation

Objectives : At the end of this lesson you shall be able to

- state the basic steps involved in troubleshooting a PC
- explain the basic approach to solve a problem
- list the probable defects and symptoms in a faulty Computer
- analyse the causes for the complaints
- state the shortest path for servicing the defects.
- list the probable defect and symptoms in the faulty Computer
- analyse the causes for the complaint “When windows is started, system runs surface test and goes to safemode” with the help of a Problem Tree and TSC.

Basic Troubleshooting

One of the difficulties while troubleshooting problems on a PC is that in most cases they are not what they seem. The cause behind a frequent hanging of a PC may be due to one of six or more well defined areas or a dozen of unidentified problems. The problem could be due to software or hardware. Even with years of experience and training, PC technicians come out with troubleshooting procedures that do not solve the real problem. For example a personal computer running windows operating system with several i/o cards connected may freeze the screen , mouse, and keyboard and take as long as 3 minutes before responding. After trying out with all the options like replacing a memory module, installing new parallel port and NIC drivers, the technician finally checks the system logs to find that a vital operating system library was corrupted and needed to be reinstalled. The issue to be mentioned here is that if the technician used a systematic approach to troubleshooting, the problem would likely have been solved much sooner. Maintaining a good troubleshooting plan certainly gives us the scope to approach the problem in a more systematic and scientific manner. A troubleshooting plan is nothing but a written check list that we use for any problem. The elements that should be included in any troubleshooting plan are as follows

- Maintenance record
- Identification of possible causes
- Identification of possible solutions
- Application and testing of solutions
- Follow-up

In the maintenance record, record the hardware installed in the PC when it is installed, all preventive maintenance activities, all software updates or additions, and all hardware installations and upgrades. Further any problems that occur and the actions you take to resolve them should be recorded . when it comes to troubleshooting a PC, with the maintenance record one can pin down a problem and devise solution for it. The first entry in such record should be a profile of the PC, which includes its configuration, operating system, and the date each component was installed.

Such tables gives an idea of the type of information one should include in the sytem configuration, as summarised below:

- The processor's make, model and speed.
- Amount of system memory(RAM) and the memory module type, size and configuration of the memory
- Hard disk size and the type of interface
- Make, model and speed of the compact disk (CD) or digital versatile disk (DVD) drive
- Memory size of the video or graphics adapter
- Make, model, type and speed of the modem
- The version number of the operating system
- A list of software applications installed on the PC
- A list of peripheral equipment attached to the PC, indicating the port to which they are attached

The maintenance record should be updated each time any maintenance work is carried over on the PC. Any time new or replacement hardware is installed, record the activity and update the system configuration. The activity entries should include

- Date of the activity or changes made to the system
- The make, model and serial number of any hardware removed or added to the PC
- The name, version and publisher of any software added to the PC
- Detailed information on any configuration changes made to the basic input/output system(BIOS) or other configuration for the new device or software.

Troubleshooting approach

For solving any problems associated with PC, first go through the maintenance record of that PC and follow a systematic procedure for isolating the problem. The standard problem solving process includes the following steps

Identify the problem: This is the most difficult part of the process. To perform this step successfully collect all the data about the problem

Identify possible causes: analyse all the symptoms of the problem and try to list all the causes in order from the most likely to the least likely.

Identify possible solutions: identify solutions for each of the causes that are identified. A possible cause could have more than one possible solution.

Analyze the possible solutions: if two solutions produce the same result, consider the one which is more economic and apply the same.

After following the above steps and on solving the problem update the maintenance record and make necessary entries into it. In some cases the problems may be very clear and the solution is very transparent and even in such cases try to follow the above mentioned steps to make it a practice to follow the systematic approach. Whenever a problem occurs with a PC, while following the steps, try to collect the information about the system by answering questions such as

- Under what circumstances this problem cropped up?
- Were there any indications in the form of beep codes/error messages or any clear symptoms?
- What softwares were active when the problem happened?
- Has it happened for the first time or occurred in the past also?
- Were there any configuration changes made during the session that required a restart that was not performed?

Optimizing the PC:

A PC which was functioning absolutely well and developed a symptom of slowing down or if it is unable to keep pace with the demands of newer software, one of the possible solutions is to consider updating or optimizing the PC to enhance its performance. Optimization steps may cost money, but many involve software you already own or software readily available on the web.

Optimizing the BIOS and Boot process:

BIOS setup configuration includes many settings in the CMOS. How quickly the system boots and performs depends on the these settings. Enabling of valuable features such as system caching or using the quick POST process are very vital for optimum performance.

Optimizing the hard disk: Windows ScanDisk and Disk Defragmenter utilities are the best tools available for optimizing the hard disk in terms of usage and access speeds. ScanDisk is used to check a disk for errors and repair them or remove unrecoverable areas of the disk from the usage tables to prevent future errors. Similarly Disk Defragmenter organizes data file fragments into a more optimized and logical format that provides for faster access times and less head movement.

Optimizing the Expansion cards: The best way to optimize I/O controllers and other expansion cards is to install them in the correct order. No harm is done even if they are installed out of order, but there is some benefit to be gained from putting them in the proper sequence. On a

Pentium system, use PCI cards and avoid ISA cards, if possible. All I/O adapters including video cards, sound cards, NICs, modems and SCSI adapters are available for the PCI expansion bus. Consult the motherboard documentation and install video card in the first PCI slot, followed by the NIC, modem, and sound card, in that order.

Optimizing the processor: One can speed up the processor in the following ways

- Replace it with a faster speed or higher level processor
- Use a utility from the processor manufacturer to apply patches or fixes to the processor's logic
- Overclock the processor

The requirement for replacing the processor with a higher level or faster processor is that your motherboard and chipset will support the new processor both logically and physically. Logically the chipset and motherboard must support the bus speed of the processor and have the supporting circuits it requires. It is often much better to replace the complete motherboard. Sometimes the manufacturer of the processor may release some utilities that will improve some aspect of processor's capabilities such as video processing, buffer handling, caching and other processor based functions. Overclocking a processor means running a processor at speeds faster than it was released to support. Most processors are capable of running at speeds higher than their nominal speeds. The nominal speed of a processor is the speed at which it has been tuned to run with a certain chipset, motherboard, cooling system, and other components of the PC. Raising the speed of the processor can create heating problems on the processor and lead to frequent system lockups, memory problems and other issues.

Troubleshooting sources of Non-software problems:

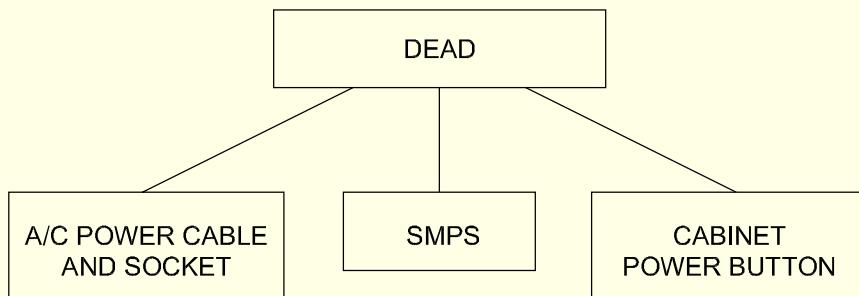
Any time pc fails for no apparent software reason, check the following areas

- Ensure proper AC power
- Scan the PC for a computer virus
- Ensure that CPU fan is spinning
- Ensure proper connections of external I/O connectors
- Reseat the expansion cards and check the power and data cables of internal devices
- Most of the boot problems are the result of a recent change, check out the BIOS setup configuration data
- To install any new hardware or software, visit manufacturer's web site for any known conflict or incompatibility
- Check for any resource conflict if any new hardware or software is installed.

The forthcoming lessons on Troubleshooting PC are provided with Problem Trees for different type of problems which a user faces normally. Each Tree with a specific problem gives scope to analyse the areas to be suspected or looked into for fixing the problem. This lesson includes a Problem Tree for a PC which is Dead with no display on monitor.

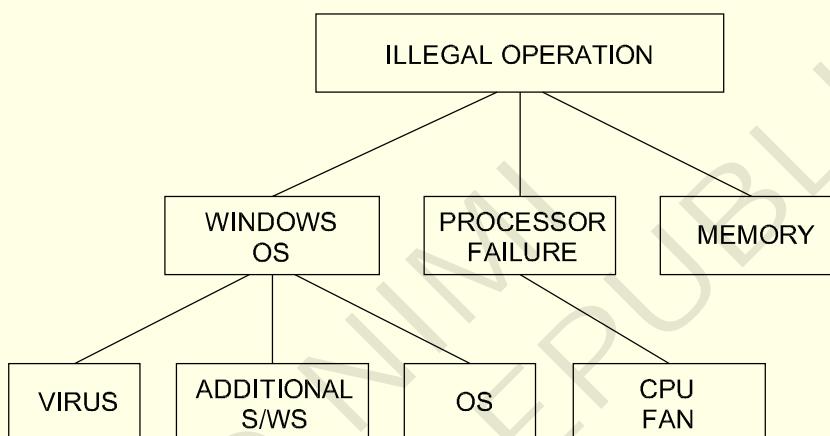
PROBLEM TREE : PT- 01

Observed symptom : Dead
 Additional symptom : No display



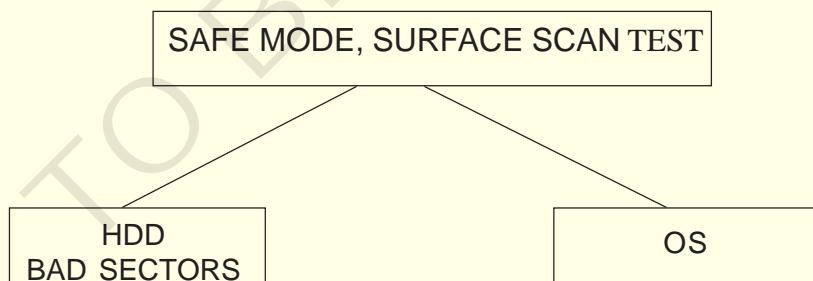
PROBLEM TREE : PT- 02

Observed symptom : Windows shows “illegal operation”
 Additional symptom : Windows not working



Discuss the Troubleshooting chart (TSC-02) and Service flow sequence (SFS-02) for the complaint “While working, windows shows error - illegal operation”.

Various faults discussed for the above complaint shall be applied to actual Computer given to you for practical exercises.

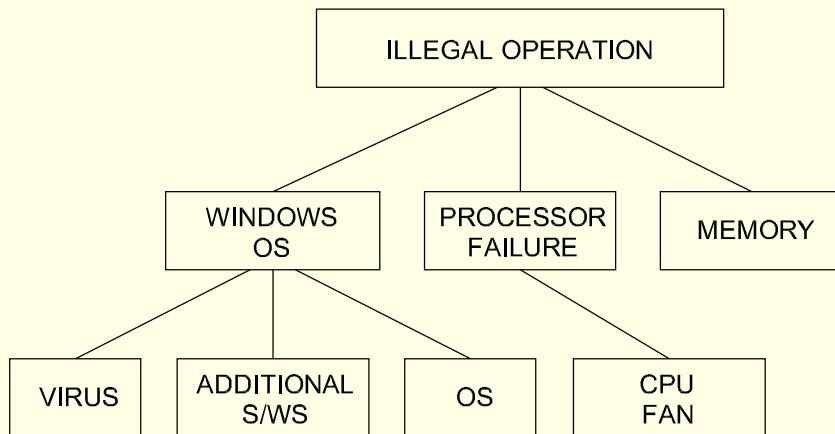


Discuss the Troubleshooting chart (TSC-01) and Service flow sequence (SFS-01) for the complaint “When windows is started system runs surface test and goes to safemode” referring to exercise 2.33

Various faults discussed for the above complaint shall be applied to actual Computer given to you for practical exercises.

PROBLEM TREE : PT- 02

Observed symptom : Windows shows “illegal operation”
Additional symptom : Windows not working

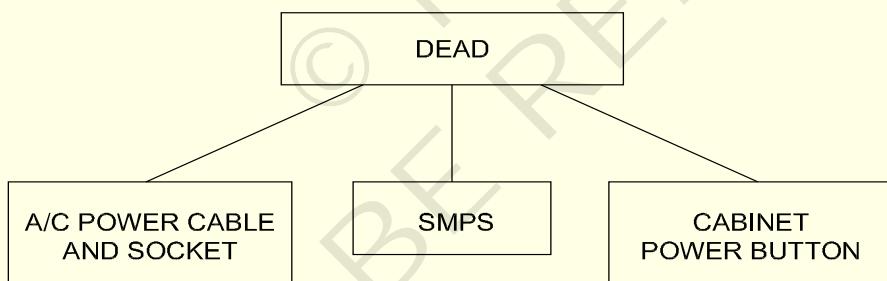


Discuss the Troubleshooting chart (TSC-02) and Service flow sequence (SFS-02) for the complaint “While working, windows shows error - illegal operation”

Various faults discussed for the above complaint shall be applied to actual Computer given to you for practical exercises.

PROBLEM TREE : PT- 03

Observed symptom : DVD drive can't read
Additional symptom :



Install necessary application software for Windows i.e. Office Package, PDF Reader, Media Player etc

Objective: At the end of this exercise you shall be able to

- describe various software types.

Application software

Application software is a term which is used for software created for a specific purpose. It is generally a program or collection of programs used by end users. It can be called an application or simply an app.

In fact all the software other than system software and programming software are application software.

Application software definition

A software which is developed to help the user to perform specific tasks is called application software.

The different types of application software include the following:

Application Software Type	Examples
Word processing software	MS Word, WordPad and Notepad
Database software	Oracle, MS Access etc
Spreadsheet software	Apple Numbers, Microsoft Excel
Multimedia software	Real Player, Media Player
Presentation Software	Microsoft Power Point, Keynotes
Enterprise Software	Customer relationship management system
Information Worker Software	Documentation tools, resource management tools
Educational Software	Dictionaries: Encarta, Britannica Mathematical: MATLAB Others: Google Earth, NASA World Wind
Simulation Software	Flight and scientific simulators
Content Access Software browsers	Accessing content through media players, web
Application Suites	OpenOffice, Microsoft Office
Software for Engineering and Product Development	IDE or Integrated Development Environments

There are various different types of application software such as licensed, sold, freeware, shareware and open source.

Application software's either need to be installed or can run online. Application software's can also be distinguished on the basis of usage into the following:

- Utility programs
- Generic programs
- Integrated programs
- Specific software
- Bespoke software
- Word processing software

- Desktop publishing software
- Spreadsheet software
- Database software
- Presentation software
- Internet Browsers
- Email Programs
- Graphic Programs (Pixel based)
- Graphic Programs (vector based)
- Communication software: Communication through audio, video or chat based means

Configure Bluetooth and Wi-Fi settings

Objective: At the end of this exercise you shall be able to

- describe the meaning of Bluetooth
- describe the method of using Bluetooth
- list the major applications of Bluetooth.

Introduction

Bluetooth is a wireless technology standard for exchanging data over short distances (using short-wavelength UHF radio waves in the range 2.4 to 2.485 GHz) from fixed and mobile devices, and building personal area networks (PANs). It is a standard wire-replacement communications protocol primarily designed for low-power consumption, with a short range based on low-cost transceiver microchips in each device.

It can connect up to eight devices (items of electronic equipment) at the same time. The chip can be plugged into items such as computers, digital cameras, mobile phones and faxes. Bluetooth is particularly convenient in certain situations - for example, when transferring files from one mobile phone to another without cables. Sending music and photos between a PC and a mobile phone is another useful application.

Because the devices use a radio (broadcast) communications system, they do not have to be in visual line of sight of each other, however a quasi optical wireless path must be viable.

Range is power-class-dependent, but effective ranges vary in practice varying from 10 to 100 m.

The name 'Bluetooth' reflects the Scandinavian origins of the technology. It is named after a 10th century Danish viking, King Harald Blåtand (translating as 'Bluetooth' in English). He united and controlled Denmark and Norway, hence the association of uniting devices through Bluetooth.

Using Bluetooth

To use Bluetooth, the device must be Bluetooth enabled. For this purpose a device called "Dongle" may be used. A dongle is a device that plugs into the computer to enable it to use Bluetooth. Every manufacturer of compatible devices will have their own instructions for accessing Bluetooth. For detailed instructions you will need to see the manual, but as a general guide:

To set up Bluetooth

(Identify the blue tooth icon on devices.)

- 1 Turn on, or enable, Bluetooth. Ensure your device is 'visible' and not 'hidden', so other nearby devices can pick up the signal.
- 2 Give your device a name to identify it when connecting to other compatible equipment.

When devices like mobile phones, laptops, tablets etc. enable Bluetooth the Bluetooth icon is on.

To establish a Bluetooth connection

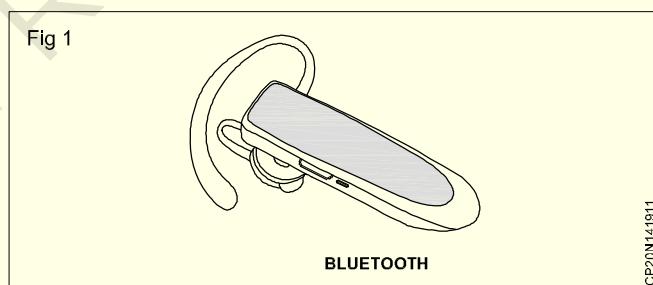
- 1 Find the file you wish to send.
- 2 Select the option to send it via Bluetooth - your device will search for other devices within range and display them.
- 3 Scroll to the device you wish to connect with and select it.
- 4 If the other device needs 'pairing', you will need to enter a passcode - a bit like a PIN number - and make sure it is entered on the other device.

When the connection is established, the data will start to send. You do not need worry about a clear line of sight between devices.

List of Bluetooth applications

Some of the Bluetooth applications are as follows:

Wireless control of and communication between a mobile phone and a handsfree headset. (Fig 1)



CP20141911

- Wireless control of and communication between a mobile phone and a Bluetooth compatible car stereo system.
- Wireless control of and communication with tablets and speakers such as iPad and Android devices.
- Wireless networking between PCs in a confined space and where little bandwidth is required.
- Wireless communication with PC input and output devices, the most common being the mouse, keyboard and printer.
- Transfer of files, contact details, calendar appointments, and reminders between devices with OBEX(Objects exchange).

- Replacement of previous wired RS-232 serial communications in test equipment, GPS receivers, medical equipment, bar code scanners, and traffic control devices.
- Wireless bridge between two Industrial Ethernet networks.
- Wireless controllers in gaming consoles.
- Personal security application on mobile phones for prevention of theft or loss of items. The protected item has a Bluetooth marker (e.g., a tag) that is in constant communication with the phone. If the connection is broken (the marker is out of range of the phone) then an alarm is raised.

Wi-Fi

Wi-Fi is a popular wireless networking technology. Wi-Fi stands for "wireless fidelity". The Wi-Fi was invented by NCR corporation/AT&T in Netherlands in 1991. By using this technology we can exchange the information between two or more devices. Wi-Fi has been developed for mobile computing devices, such has laptops, but it is now extensively using for mobile applications and consumer electronics like televisions, DVD players and digital cameras. There should be two possibilities in communicating with the Wi-Fi connection that may be through access point to the client connection or client to client connection. Wi-Fi is a one type of wireless technology. It is commonly called as wireless LAN (local area network). Wi-Fi allows local area networks to operate without cable and wiring. It is making popular choice for home and business networks. A computer's wireless adaptor transfers the data into a radio signal and transfers the data into antenna for users. (Fig 2)

Fig 2



Copa 1121801

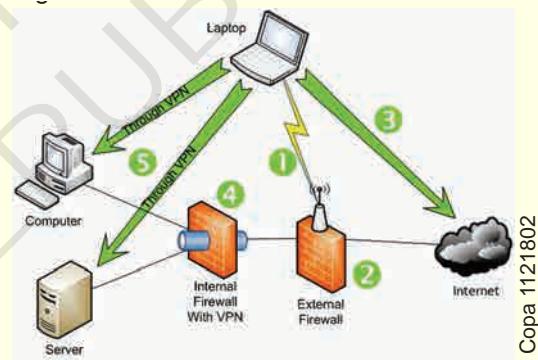
Working Principle

Wi-Fi is a high speed internet connection and network connection without use of any cables or wires. The wireless network is operating three essential elements that are radio signals, antenna and router. The radio waves are keys which make the Wi-Fi networking possible. The computers and cell phones are ready with Wi-Fi cards. Wi-Fi compatibility has been using a new creation to constituent within the ground connected with community network. The actual broadcast is connected with in sequence in fact it

is completed by way of stereo system surf as well as the worth of wires with monitor to classification prone. Wi-Fi allows the person in order to get access to web any place in the actual provided area. And can now generate a system within Resorts, library, schools, colleges, campus, personal institutes, as well as espresso stores as well as on the open public spot to help to make the company much more lucrative as well as interact with their own customer whenever. Wi-Fi compatibility can make surf with stare to company using their inspiring cable television much a smaller amount force down.

The radio signals are transmitted from antennas and routers that signals are picked up by Wi-Fi receivers, such has computers and cell phones that are ready with Wi-Fi cards. Whenever the computer receives the signals within the range of 100-150 feet for router it connect the device immediately. The range of the Wi-Fi is depends upon the environment, indoor or outdoor ranges. The Wi-Fi cards will read the signals and create an internet connection between user and network. The speed of the device using Wi-Fi connection increases as the computer gets closer to the main source and speed is decreases computer gets further away. (Fig 3)

Fig 3



Copa 1121802

Security:

Security is impartment element in the Wi-Fi technology. Security is our personal decision but having a wireless connection we should pay attention to protect our private details. We can connect easily to unsecured wireless routers. The problem is any one is connected to the wireless router using the data like download games, download apps and planning terrorist activities, shirring illegal music and movie files etc. So it is necessary to provide security to the wireless technologies based devices.

DVDs, CDs and burning DVDs

Objectives : At the end of this lesson you shall be able to

- describe the features of CDs & DVDS
- describe the main formats of DVDs
- describe DVD burning
- describe the features of CDs.

Introduction

DVD (sometimes called as "digital video disc" or "digital versatile disc") is a digital optical disc storage format. DVDs can be used with many types of players, including PCs and standalone players.

These discs are known as DVD-ROM, because data can only be read and not written or erased. Blank recordable DVD discs (DVD-R and DVD+R) can be recorded once using a DVD recorder and then function as a DVD-ROM. Rewritable DVDs (DVD-RW, DVD+RW, and DVD-RAM) can be recorded and erased multiple times.

DVD features and formats

DVDs are used in DVD-Video consumer digital video format and in DVD-Audio consumer digital audio format. They can also be used in a special AVCHD format (Advanced Video Coding High Definition) often used with AVCHD format camcorders. DVDs containing other types of information may be referred to as DVD data discs.

A typical recordable DVD can hold about 4.7 gigabytes (GB). However, the total amount of disc space that you can use to burn files to the disc is less than the amount that's often listed on the disc itself. This is because the disc capacity is calculated differently when it's used in a computer. For example, a typical DVD-R that has a listed disc capacity of 4.7 GB can only store about 4.37 GB of data on the disc. DVDs offer a storage capacity of approximately 4.7 GB. DVD discs do not deteriorate over time and are unaffected by magnetic fields.

The type of recordable disc to be used depends on a few different factors, such as:

- The types of recordable discs that work with the disc burner.
- The disc drive on the computer or device will read the disc after it is burned.
- The total size of all the files that will be burned on to a disc.

DVD burning: The process of recording source material onto an optical disc (CD or DVD) is called burning / writing or optical disc authoring. Creating an optical disc usually involves first creating a disk image with a full file system designed for the optical disc, and then actually burning the image to the disc. Many programs are available as bundled applications to create the disk image and burn the files.

The speed at which a DVD can be written is expressed as a multiplier: 16X means 16 times faster than just playing it. Speeds up to 52X are also very common.

CD or DVD formats

For burning DVDs, the two main disc formats in use are:

- Live File System and
- Mastered disc formats.

Live File System format

Discs that use the Live File System format are often more convenient because you can copy selected files immediately and as often as you want, just like you can with a USB flash drive. This is convenient if you want to keep a disc in your CD, DVD, or Blu ray Disc burner and copy files whenever the need arises. In this format you can copy and erase files over and over again. However, the Live File System optical disc format is only readable by Windows 7, Windows Vista, and Windows XP systems. These discs are not blank after they're formatted.

Discs formatted in this format have the option name in the Burn a Disc dialog box: "Like a USB flash drive."

Mastered disc formats

If we want to create an optical disc that can be used to transfer data files to older versions of the Windows operating system or even to another operating system, we need to use the Mastered optical disc format. Also if we want to burn music or pictures and use the disc in regular CD, DVD, or Blu ray Disc players that can play MP3 files and show digital pictures, we should use the Mastered optical disc format.

The Mastered format works just like burning CDs in Windows XP. In other words, when we write the disc, we copy a file or a group of files to the optical disc all at once. Once this is done, the disc is closed and we cannot copy more files to the disc nor can we delete the existing files. Hence it is recommended not to copy files immediately; it is a good practice to assemble the entire collection of files that needed to be copied to the disc and then burn them all at once.

Discs formatted with the Mastered option have the option in the Burn a Disc dialog box: "With a CD/DVD player."

There are many types of tools available to create data, music, video and audio discs. We can also create backups that span across multiple discs, rip music tracks from Audio CDs and create or burn disc images in different formats. They may also provide features like automatic audio conversion from WAV, MP3, FLAC, WMA files, disc copying, compressed file backup and restore, disk erasing, VCD/SVCD support, project burning etc..

Blu - ray Discs

Blu-ray, also known as Blu-ray Disc (BD) is the name of a new optical disc format that is rapidly replacing DVD. The format was developed to enable recording, rewriting and playback of high-definition video (HD), as well as storing large amounts of data. The format offers more than five times the storage capacity of traditional DVDs and can hold up to 25GB on a single-layer disc and 50GB on a dual-layer disc.

The name Blu-ray is derived from the underlying technology, which utilizes a blue-violet laser to read and write data. The name is a combination of "Blue" (blue-violet laser) and "Ray" (optical ray). They are referred to as "Blu-ray" discs or BDs.

The following formats are part of the present day Blu-ray Disc specification:

- 1 BD-ROM - read-only format for distribution of HD movies, games, software, etc.
- 2 BD-R - recordable format for HD video recording and PC data storage.
- 3 BD-RE - rewritable format for HD video recording and PC data storage.

At present, a single-layer disc can hold 25GB and a dual-layer disc can hold 50GB. Over 9 hours of high-definition (HD) video on a 50GB disc. About 23 hours of standard-definition (SD) video on a 50GB disc.

Use basic DOS commands for directory listing

Objectives: At the end of this lesson you shall be able to

- describe the hierarchical directory system in DOS
- use dos commands to create directories and subdirectories
- use dos commands to change and list directory
- use dos commands to access specific files.

Hierarchical Directory System: Hierarchy in simple terms, is, organisation or an arrangement of entities. Entities can be anything such as objects, files, people, ideas, or any other thing.

Arrangement refers to, for example, Currency can be arranged by denomination. Pebbles can be arranged by their size .

There are many other ways to organize entities besides hierarchically. But, hierarchical organization is special because by this arrangement you can name each entity by its relationship to other entities.

In DOS, entities are the *Directories* in a directory system. Here, the hierarchy begins with the essential *core* or *root entity*. For instance, in a family tree, we may consider great-great-grand father who was the root cause of our existence as the core entity. In DOS , this core entity is referred to as the the *root directory*.

As in the example considered above, if we consider great-great-grandfather as the *root directory*, then, great-grand father, grand father, father are referred as *sub directories*. So the directories under the root directory are called subdirectories in DOS. These subdirectories can trace their paths back to the root directory.

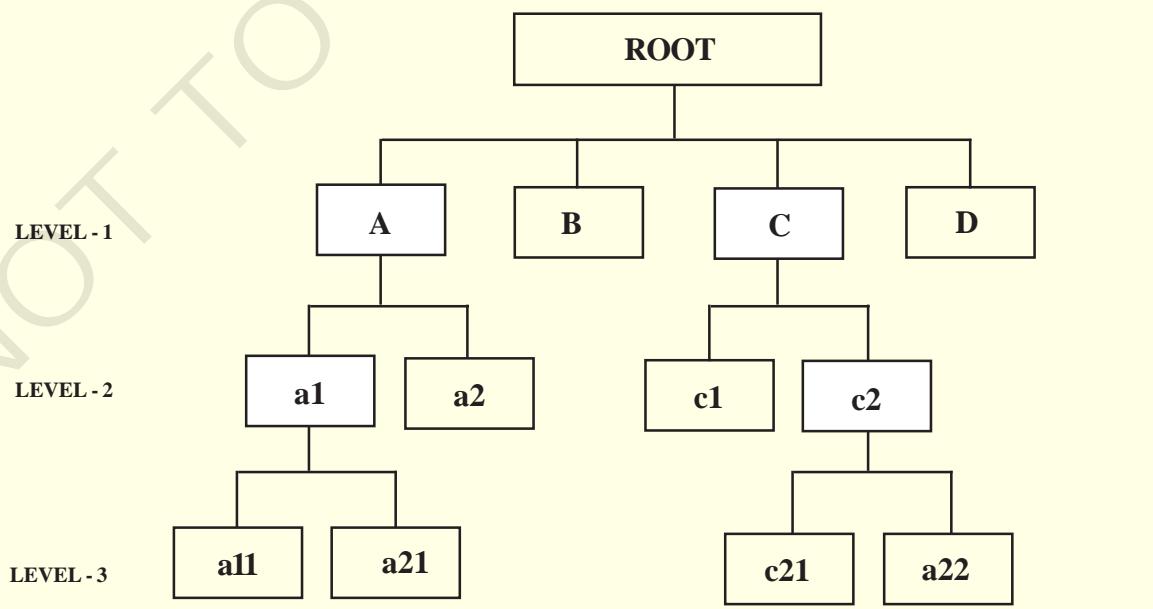
The DOS hierarchical file system is called a *tree-structured file system*. At the base of this tree structure is the root directory.

In a family tree, say, Govinda is the son of Rajappa, who is son of Ramappa who is son of Venkappa. Venkappa is the head or root of the family tree for Govinda.

One can create many directories from the root. The root will then be the parent of each of these directories. You can also create subdirectories that stem from other subdirectories that stem from other subdirectories and so on. These new subdirectories have a subdirectory as their parent directory. How subdirectories are arranged hierarchically from the root is illustrated in Fig 1. The DOS directory system is often called a *tree-structured directory system*.

Three levels of subdirectories are represented in Fig 1. Regardless of the number of levels, the relationship of the subdirectories is important. Each subdirectory, as well as the root directory, can contain user files. Two files can have the same file name and extension as long as the files reside in different directories. This is because, DOS needs to know which of two same-named files your command specifies. For this, DOS needs the name of the directories, starting from the root, that lead to the desired file. This sequence of directory names leading to a file is called a **path**.

Fig 1



A path is a chain of directory names that tell DOS how to find a file that you want. Each directory is separated from the other by a ‘\’ character. This ‘\’ is referred to as the DOS *directory delimiter*. A files full path name including the drive specifier (C: or D: etc.,) is the absolute indicator of where the file is located. Typical path notation are given below;

D:\Animals\Domestic\Pets\Dog.txt

C:\Admin\Accounts\Tours\Bata.txt

Further details of path and directory structure will be discussed at appropriate lessons.

DOS COMMANDS

1 MKDIR Makes or Creates a new Directory.

or

MD

Syntax

MKDIR C:path\dirname

Or

MD d:path\dirname

Where,

C: is the disk drive for the sub directory

path\ indicates the path to the directory that will hold the subdirectory being created.

dirname is the name of the subdirectory being creating.

Switch

(None)

Important Notes

- **MKDIR or its short form MD makes new subdirectories under the selected root directory.**
- It is possible to create as many subdirectories as you want , but remember: *DOS accepts no more than 63 characters, including backslashes, for the path name.*
- Do not create too many levels of subdirectories and with long names.
- You cannot create a directory name that is identical to a file name in the current directory.

For example, if you have a file named FLIES in the current directory, you cannot create a subdirectory by the name FLIES in this directory. However, if the file FLIES is given an extension FLIES.DOC , then the names will not conflict and you can create a subdirectory by name FLIES.

Examples

To create the subdirectory by name **Drivers** under the **current drive**, the instruction will be,

MKDIR\Drivers

Or

MD\Drivers

C:\Devices>MD \Printers

This instruction creates a subdirectory by name **Printers** under the current drive C:. Note that although the command is issued from another subdirectory named devices, the newly created subdirectory **Printers** does not get created under the directory Devices but directly under the root C:. This may be verified by issuing DIR command under C:\ and under C:\Devices.

To create a subdirectory under the directory Devices the instruction will be,

C:\Devices>MD Printers

Discuss the following different varieties of creating directories:

C:\Devices\Printers>MD C:\Devices\Plotters

2 CHIDR or CD

Changes or shows the path of the current directory.

Syntax

CHIDR d: path

Or, using the short form:

CD d : path

D : path are valid disk drive and directory names.

Switch

(None)

You have two methods for maneuvering through the hierarchical directories with CD: (1) starting at the root , or top, directory of the disk and moving down, or (2) starting with the current directory and moving in either direction.

To start at the root directory of a disk, you must begin the path with the path character (\), as in \ or B:\. When DOS sees \ as the first character in the path, the system starts with the root directory. Otherwise, DOS starts with the current directory.

Changing Drives: Computer will have built in memory, the hard disk and it will also have provision to store/read data from floppy disk, compact disk etc. Every disk is identified by a name such as C drive, A drive, B drive etc. C drive is represented by C: and A drive is represented by A: and so on. DOS allows to change from current or default drive by typing the letter identification of disk drive desired followed immediately by a colon as shown in the example below:

C> a:

This command instructs to change control from **C** drive to **D** drive.

If the disk drive is not accessed due to non availability of floppy or any other reason, DOS will display an error message

Not ready error reading drive A

Abort, Retry, Fail ?

It is required to press either A,R or F keys, which are defined below

A Directs DOS to abort the command that was responsible for the error. If this option is selected DOS will terminate the command and redisplay prompt.

R Directs DOS to retry the command that caused the error. In most cases this option is selected to correct the problem that was causing the error. (Floppy disk might not be inserted).

F Directs DOS to ignore the error and attempt to continue processing. In some cases DOS will have an error when it reads a portion of disk.

DOS COMMAND

DIR Displays a list of files and subdirectories in a directory.

Syntax

```
DIR C:path/filename [/P] [/W] [/A[[:attribs]]] [/O[[:sortord]]]  
/[S] [/B] [/L] [/C[H]]
```

Where,

- **C:** is the disk drive holding the directory for displaying list of files and subdirectories
- **path/** specifies directory and/or files to list.
- **filename** specifies file or list of files to display, if file name is not specified all the files in the directory will be listed.
- **[/P] [W]** specifies the switches for formatting the output.

Switch

/P Pauses after each screenful of information and waits to press any key. On pressing any key another screenful or remaining information will be displayed. Command is DIR/P

/W Uses wide format of 80-column to display file names only and information about file size, date, and time is not displayed. Command is DIR/W

/A Displays files with specified attributes.

attribs

D Directories

R Read-only files

H Hidden files

S System files

A Files ready to archive - Prefix meaning "not"

/O List be files in sorted order.

sorted N By name (alphabetic)

S By size (smallest first)

E By extension (alphabetic)

D By date & time (earliest first)

G Group directories first

- Prefix to reverse order

C By compression ratio (smallest first)

/S Displays files in specified directory and all subdirectories.

/B Uses bare format (no heading information or summary).

/L Uses lowercase.

/C[H] Displays file compression ratio; /CH uses host allocation unit size.

Important Notes:

- In the directory listing similar files can be listed by using wildcards (* and ?), where (*) star and (?) question mark are called wild characters or wild cards. * can replace remaining characters and ? can replace any single character.
- When DIR is used without parameters or switches, It displays the disks volume label and serial number; one directory or filename per line, including the file size in bytes, and the date and time the file was modified; and the total number of files listed, their cumulative size and the free space (in bytes) remaining on the disk.

Examples

DIR *.txt

*.txt instruction will list all files having txt extension in the specified directory.

DIR ???T.*

???T instruction will search for files having four characters which ends with T like TEST, REST etc. And * instructs that these files may have any extension like .txt, .dat etc.

Manage files and folders using DOS commands

Objectives: At the end of this lesson you shall be able to

- use DOS commands to display the contents of a text file
- use DOS commands to copy, rename, delete and undelete files.

DOS Commands

TYPE Displays the contents of a text file.

Syntax

TYPE C: path/filename

Where,

- **C:** is the disk drive holding the file for displaying.
- **path/** Specifies the location of file for displaying.
- **filename** specifies file to display.

Switch

(none)

Important notes:

- **TYPE** command provides a quick way to display contents of an ASCII file without having to use another program. The file is stored on the disk as ASCII (American Standard code for Information Interchange) text which is standard way the computer translates binary (ones and zeros) into letters, numbers & symbols. If the information is not stored in the ASCII format, on using TYPE command the information will look like gibberish.
- On issuing command DOS will look in drive specified, moves into the path to reach the filename specified. Then it simply translates ASCII format into the characters, numbers and symbols and displays on the monitor. The video monitor can show 24 lines of information only. If the file contains more than 24 lines starting lines can not be seen since the type command simply scrolls all information on to the screen. Scrolling can be controlled by pressing Control + S keys together (on holding control key press S key and release both the keys is called as Control + S) scrolling of information will stop on the monitor. After viewing the contents on the screen any key can be pressed to scroll through the remaining contents. To view the contents of the file screen page by screen page, MORE command can also be used which will stop the scrolling of information on the screen exactly after a screen page and in the screen page at 24 line a prompt message — More — is displayed. After pressing any key another screen page will be displayed. MORE is a filter e.g. it is a program that manipulates the stream of standard characters to the file to the standard output (monitor) screen page by screen page.

Examples

- 1 C:\COPA\ DOS\ PRACT_3>TYPE TEST1.txt
C:\COPA\ DOS\ PRACT_3 is the path to the file TEST.txt and TYPE is the command to be executed by DOS.
- 2 C:\>TYPE C:\COPA\ DOS\ PRACT_3>TEST1.txt
This results in the same output as in example1. While working from C: (C drive) this command can be issued without changing the directories.
- 3 C:\COPA\ DOS\ PRACT_3>TYPE TEST1.txt | MORE
This will also result in the same output but displayed screen page by screen page. Screen page can be changed on press of any key. Along with MORE another character is prefixed ‘|’ this is called the piping command, Which will route the output of TYPE command to another command MORE and the MORE filter outputs the information.

Renaming of file(s)

RENAME This command allows to change
Or the name of a file.

REN

Syntax

REN C: PATH\filename1.ex1 filename2.ex2

Where,

- **C:** is the disk drive holding the file for displaying.
- **PATH/** Specifies the location of file for displaying.
- **filename1.ex1** is the file to be renamed
- **filename2.ex2** is the new filename

Important Notes

- If the drive is not specified current disk drive will be used.
- If the path is not specified current directory will be used
- Exact file name with extension is to be given for the file to be renamed.
- A valid file name with appropriate extension is to be given for new filename.
- Wild characters are permitted in the file names by which required group of files can be renamed.

- Only file names will be changed and contents remain same.
- If attempted to change a file name to a name that already exists in the directory.

DOS prompts an error message

Duplicate file name or file not found

- If a invalid file name or the new name is not given, then also DOS prompts an error message

Rules for the file names.

- A File name must have 1 to 8 characters.
- An optional extension of 1 to 3 characters
- A period (.) between the name and extension name, if extension is used
- All letters from A through Z (lower case letters are automatically transferred to uppercase), 0 to 9 numbers and special characters & symbols \$ # & @ ! ^ () _ - { } ' ~ are permitted in the file name.
- The control characters such as Esc, Del, or space bar cannot be used in the file name.
- The characters + = / [] : ; ? * < > : are not permitted.
- Each file name in a directory must be unique.

Examples:

1 C:\COPA\ DOS\ PRACT_3>REN TEST2.txt
CHECKED.txt

C:\COPA\ DOS\ PRACT_3 is the drive and path to the TEST2.txt file

TEST2.txt is the file name to be renamed

CHECKED.txt is the new filename

2 C:\COPA\ DOS\ PRACT_3>REN *.pic *.jpg the pic extension will be changed to jpg extension files.

Copying files:

COPY Copies one or more files to another location.

Syntax

COPY [/A | /B] source [/A | /B] [+ source [/A | /B] [+ ...]] [destination[/A | /B]] [/V] [/Y | /-Y] source specifies the file or files to be copied. Destination specifies the directory and/or filename for the new file(s).

Switches

- /A Indicates an ASCII text file.
- /B Indicates a binary file.
- /V Verifies that new files are written correctly.
- /Y Suppresses prompting to confirm you want to overwrite an existing destination file.
- /-Y Causes prompting to confirm you want to overwrite an existing destination file.

Instructor shall discuss the simple switches with at least two examples in each case .

For further details on COPY command switches refer any tutorial or hand book on DOS

Important Notes:

- DOS command COPY can duplicate one or more files. In the same directory with different names or from one directory to other directory either in the same name or in different name.
- If the drive is not specified current disk drive will be used.
- If the path is not specified current directory will be used.
- Exact file name with extension is to be given for the file to be copied
- A valid file name with appropriate extension is to be given for new copied filename
- Wild characters are permitted in the file names by which required group of files can be copied
- On copying, both source and target files will have same contents.
- Copy overwrites the target file with the same name
- Copy will not allow to copy a file to it self that is source and target files should not be same
- If the destination file name is not specified while concatenation the first file name will become the destination name. After the first file name, additional source files must be preceded by a plus (+) sign.

Example

1 C:\COPA\ DOS\ PRACT_3>**COPY TEST2.txt TRIAL.txt**

With the above command C:\COPA\ DOS\ PRACT_3 directory TEST2.txt file will be copied as TRIAL.txt file in the same directory. On listing the directory both the files will have same details and on viewing the contents of both the file will be same. After copying DOS prompts a message 1 file copied

2 C:\COPA\ DOS\ PRACT_3>**COPY *.bmp *.pic**

With the above command C:\COPA\ DOS\ PRACT_3 directory all files with bmp extension file will be copied as pic extension files in the same directory. While copying DOS prompts the name of file it has copied and after completion of copying it prompts the number of files copied.

3 C:\COPA\ DOS\ PRACT_3>**COPY *.pic C:\COPA\ DOS\ PRACT_4**

All files with pic extension in C:\COPA\ DOS\ PRACT_3 directory will be copied to C:\COPA\ DOS\ PRACT_4 directory with same name & extension.

Using *.* after the copy command will copy all files with all extension to the destination.

Copy concatenating: Multiple file can be combined to form a single file by use of + between the source files and is called as concatenation

Example 4 C:\COPA\ DOS\ PRACT_3\ COPY TEST2.txt
+ TRIAL.txt CONCAT1.txt

With the above command TEST2.txt and TRIAL.txt will be combined and CONCAT1.txt file will be created which will have the contents of first two source files.

Deleting file

DEL Deletes the files specified.

or

Erase

Syntax

DEL C: path/filename [/P]

ERASE C: path/filename [/P]

Where,

- **C:** is the disk drive holding the file to be deleted.
- **path/** Specifies the location of file to be deleted.
- **filename** is the file to be deleted

Switch

/P Prompts for confirmation before deleting the specified file. Using the /P switch

If the /P switch is used, DEL displays the name of a file and prompts with a message in the following format:

filename, Delete (Y/N)?

Press Y to confirm the deletion, N to cancel the deletion and display the next filename (if a group of files are specified), or CRTL+C to stop the DEL command.

Important Notes

- If the drive is not specified current disk drive will be used
- If the path is not specified current directory will be used
- Exact file name with extension is to be given for the file to be deleted
- Wild characters are permitted in the file names by which required group of files can be deleted
- On deleting, files name(s) will be removed from the directory.
- All the files in a directory can be deleted by typing the DEL command followed by [drive:]path. Wildcard also can be used (*) and (?) to delete more than one file at a time. However, Wildcards should be used cautiously with the DEL command to avoid deleting files unintentionally.

The following command is given for deleting all the files.

del *.*

DEL displays the following prompt:

All files in directory will be deleted! Are you sure (Y/N)?

Press Y and then ENTER to delete all files in the current directory, or

press N and then ENTER to cancel the deletion.

- Directories can not be removed with DEL command a separate command is available for removing the directory.
- Once the file is deleted it can not be recovered if the memory space is occupied by a new file. If accidentally file (s) are deleted immediately it can be recovered by using utility command.
- The space occupied by the deleted file on the disk or diskette is freed.
- Check for the typographic errors in the file names before the press of enter key to activate delete command

Example

1 C:\COPA\ DOS\ PRACT_3\>DEL TEST2.txt

With the above command TEST2.txt file will be deleted from the C:\COPA\ DOS\ PRACT_3 directory. On listing the directory TEST2.txt will not be available.

2 C:\COPA\ DOS\ PRACT_4\>DEL *.txt

With the above command in the C:\COPA\ DOS\ PRACT_4 directory all files with txt extension will be deleted.

3 C:\COPA\ DOS\ PRACT_3\TEMP > DEL *.*

All files with any extension in C:\COPA\ DOS\ PRACT_3\TEMP directory will be deleted.

Recovering deleted files:

UNDELETE delete protection facility

Syntax

UNDELETE C: path/filename [/DT | /DS | /DOS]

UNDELETE [/LIST | /ALL | /PURGE[DRIVE] | / STATUS | /LOAD | /UNLOAD

/UNLOAD | /S[DRIVE] | /T[DRIVE]-entries]]

Where,

- **C:** is the disk drive holding the files to be undeleted.
- **path/** Specifies the location of file to be undeleted.
- **filename** is the file to be undeleted

Switches

/LIST	Lists the deleted files available to be recovered.
/ALL	Recovering files without prompting for confirmation.
/DOS	Recovering files listed as deleted by MS-DOS.
/DT	Recovering files protected by Delete Tracker.
/DS	Recovering files protected by Delete Sentry.
/LOAD	Loads Undelete into memory for delete protection.
/UNLOAD	Unloads Undelete from memory.
/PURGE[drive]	Purges all files in the Delete Sentry directory.
/STATUS	Display the protection method in effect for each drive.
/S[drive]	Enables Delete Sentry method of protection.
/T[drive][-entries]	Enables Delete Tracking method of protection.

Important Notes

Once a file is deleted from disk, it may not be possible to retrieve it. Although the UNDELETE command can retrieve deleted files, it can do so with certainty only if no other files have been created or changed on the disk. If a file is accidentally deleted and it is required to keep, stop what all other activities on the computer and immediately use the UNDELETE command to retrieve the file.

Example

- 1 C:\COPA\ DOS\ PRACT_3>UNDELETE TEST2.txt
With the above command TEST2.txt file will be recovered. On listing TEST2.txt file will be available in C:\COPA\ DOS\ PRACT_3 directory.
- 2 C:\COPA\ DOS\ PRACT_4\TEMP>UNDELETE
With the above command multiple files can be recovered. DOS will prompt for confirmation of undeletion of each file and asks to type the first letter of the file. After undeletion and listing of C:\COPA\ DOS\ PRACT_4 directory, undeleted file names can be seen .
- 3 C:\COPA\ DOS\ PRACT_4\TEMP>UNDELETE /ALL
With the above command multiple files can be recovered. DOS will not prompt for confirmation of undeletion of each file. After undeletion and listing of C:\COPA\ DOS\ PRACT_4 directory, undeleted file names can be seen.

Introduction to Linux operating system

Objectives: At the end of this lesson you shall be able to

- **overview of linux**
- **define futures of linux**
- **explain application area of linux**
- **describe about kernel.**

Overview of Linux

The operating system

Developers need special tools (like the compilers and command lines found in GNU) to write applications that can talk to the kernel. They also need tools and applications to make it easy for outside applications to access the kernel after the application is written and installed.

This collective set of tools, combined with a kernel, is known as the operating system. It is generally the lowest layer of the computer's software that is accessible by the average user. General users get to the operating system when they access the command line.

Linux provides powerful tools with which to write their applications: developer environments, editors, and compilers are designed to take a developer's code and convert it to something that can access the kernel and get tasks done.

Like the kernel, the Linux operating system is also modular. Developers can pick and choose the operating tools to provide users and developers with a new flavor of Linux designed to meet specific tasks.

Introduction to Linux

Linux (pronounced Lih-nucks) is a UNIX-like operating system that runs on many different computers. Although many people might refer to Linux as the operating system and included software, strictly speaking, Linux is the operating system kernel, which comes with a distribution of software.

Linux was first released in 1991 by its author Linus Torvalds at the University of Helsinki. Since then it has grown tremendously in popularity as programmers around the world embraced his project of building a free operating system, adding features, and fixing problems.

Linux is popular with today's generation of computer users for the same reasons early versions of the UNIX operating system enticed fans more than 20 years ago. Linux is portable, which means you'll find versions running on name-brand or clone PCs, Apple Macintoshes, Sun workstations, or Digital Equipment Corporation Alpha-based computers. Linux also comes with source code, so you can change or customize the software to adapt to your needs. Finally, Linux is a great operating system, rich in features adopted from other versions of UNIX.

Where is Linux?

One of the most noted properties of Linux is where it can be used. Windows and OS X are predominantly found on personal computing devices such as desktop and laptop computers. Other operating systems, such as Symbian, are found on small devices such as phones and PDAs, while mainframes and supercomputers found in major academic and corporate labs use specialized operating systems such as AS/400 and the Cray OS.

Linux, which began its existence as a server OS and has become useful as a desktop OS, can also be used on all of these devices. „From wristwatches to supercomputers,“ is the popular description of Linux' capabilities.

The future of Linux

Linux is already successful on many different kinds of devices, but there are also many technological areas where Linux is moving towards, even as desktop and server development continues to grow faster than any other operating system today.

Linux is being installed on the system BIOS of laptop and notebook computers, which will enable users to turn their devices on in a matter of seconds, bringing up a streamlined Linux environment. This environment will have Internet connectivity tools such as a web browser and an e-mail client, allowing users to work on the Internet without having to boot all the way into their device's primary operating system—even if that operating system is Windows.

At the same time, Linux is showing up on mobile Internet devices (MIDs). This includes embedded devices such as smart phones and PDAs, as well as netbook devices—small laptop-type machines that feature the core functionality of their larger counterparts in a smaller, more energy-efficient package.

The growth of cloud computing is a natural fit for Linux, which already runs many of the Internet's web servers. Linux enables cloud services such as Amazon's A3 to work with superior capability to deliver online applications and information to users.

Related to Linux' growth in cloud computing is the well-known success of Linux on supercomputers, both in the high-performance computing (HPC) and high-availability (HA) areas, where academic research in physics and bioengineering, and firms in the financial and energy

industries need reliable and scalable computing power to accomplish their goals.

Many of the popular Web 2.0 services on the Internet, such as Twitter, Linked In, YouTube, and Google all rely on Linux as their operating system. As new web services arrive in the future, Linux will increasingly be the platform that drives these new technologies.

Current application of Linux operating systems

Today Linux has joined the desktop market. Linux developers concentrated on networking and services in the beginning, and office applications have been the last barrier to be taken down. They don't like to admit that Microsoft is ruling this market, so plenty of alternatives have been started over the last couple of years to make Linux an acceptable choice as a workstation, providing an easy user interface and MS compatible office applications like word processors, spreadsheets, presentations and the like. On the server side, Linux is well-known as a stable and reliable platform, providing database and trading services for companies like Amazon, the well-known online bookshop, US Post Office, the German army and many others. Especially Internet providers and Internet service providers have grown fond of Linux as firewall, proxy- and web server, and you will find a Linux box within reach of every UNIX system administrator who appreciates a comfortable management station. In post offices, they are the nerve centres that route mail and in large search engine, clusters are used to perform internet searches. These are only a few of the thousands of heavy-duty jobs that Linux is performing day-to-day across the world. It is also worth to note that modern Linux not only runs on workstations, mid- and high-end servers, but also on "gadgets" like PDA's, mobiles, a shipload of embedded applications and even on experimental wristwatches. This makes Linux the only operating system in the world covering such a wide range of hardware.

The code

Linux is also unique from other operating systems in that it has no single owner. Torvalds still manages the development of the Linux kernel, but commercial and private developers contribute other software to make the whole Linux operating system.

The kernel

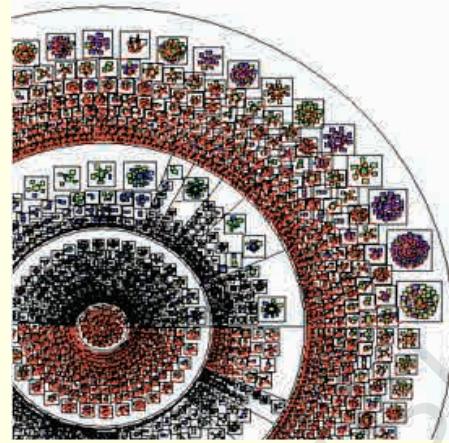
All operating systems have kernels, built around the architectural metaphor that there must be a central set of instructions to direct device hardware, surrounded by various modular layers of functionality. The Linux kernel is unique and flexible because it is also modular in nature.

Modularity is desirable because it allows developers to shed parts of the kernel they don't need to use. Typically a smaller kernel is a faster kernel, because it isn't running processes it does not need.

If a device developer wants a version of Linux to run on a cell phone, she does not need the kernel functionality that deals with disk drives, Ethernet devices, or big monitor screens. She can pull out those pieces (and others), leaving just the optimized kernel to use for the phone.

The kernel of the Window operating system (which few people outside of Microsoft are allowed to look at without paying for the privilege) is a solidly connected piece of code, unable to be easily broken up into pieces. It is difficult (if not impossible) to pare down the Windows kernel to fit on a phone.

Fig 1



This modularity is significant to the success of Linux. The ability to scale down (or up) to meet the needs of a specific platform is a big advantage over other operating systems constrained to just a few possible platforms.

Modularity also effects stability and security as well. If one piece of the kernel code happens to fail, the rest of the kernel will not crash. Similarly, an illicit attack on one part of the kernel (or the rest of the operating system) might hamper that part of the code, but should not compromise the security of the whole device.

The environments

The windows, menus, and dialog boxes most people think of as part of the operating system are actually separate layers, known as the windowing system and the desktop environment.

These layers provide the human-oriented graphical user interface (GUI) that enables users to easily work with applications in the operating system and third-party applications to be installed on the operating system.

In Linux, there a lot of choices for which windowing system and desktop environment can be used, something that Linux allows users to decide. This cannot be done in Windows and it's difficult to do in OS X.

Like the operating system and kernel, there are tools and code libraries available that let application developers to more readily work with these environments (e.g., gtk+ for GNOME, Qt for KDE).

The applications

Operating systems have two kinds of applications: those that are essential components of the operating system itself, and those that users will install later. Closed operating systems, like Windows and OS X, will not let users (or developers) pick and choose the essential component applications they can use. Windows developers must use Microsoft's compiler, windowing system, and so on.

Linux application developers have a larger set of choices to develop their application. This allows more flexibility to build an application, but it does mean a developer will need to decide which Linux components to use.

The distributions

A Linux distribution is a collection of (usually open source) software on top of a Linux kernel. A distribution (or short, distro) can bundle server software, system management tools, documentation and many desktop applications in a central secure software repository. A distro aims to provide a common look and feel, secure and easy software management and often a specific operational purpose.

Let's take a look at some popular distributions.

Red hat

Red Hat is a billion dollar commercial Linux company that puts a lot of effort in developing Linux. They have hundreds of Linux specialists and are known for their excellent support. They give their products (Red Hat Enterprise Linux and Fedora) away for free. While Red Hat Enterprise Linux (RHEL) is well tested before release and supported for up to seven years after release, Fedora is a distro with faster updates but without support.

Introduction to Open source software

Objectives: At the end of this lesson you shall be able to

- **definition of OSS**
 - **history of OSS**
 - **open source initiative**
 - **examples of OSS.**
-

Definition

Open source software (OSS) is software that is distributed with its source code, making it available for use, modification, and distribution with its original rights. Source code is the part of software that most computer users don't ever see; it's the code computer programmers manipulate to control how a program or application behaves. Programmers who have access to source code can change a program by adding to it, changing it, or fixing parts of it that aren't working properly. OSS typically includes a license that allows programmers to modify the software to best fit their needs and control how the software can be distributed.

What is the history of OSS?

The idea of making source code freely available originated in 1983 from an ideological movement informally founded by Richard Stallman, a programmer at MIT. Stallman believed that software should be accessible to programmers so they could modify it as they wished, with the goal of understanding it, learning about it, and improving it.i Stallman began releasing free code under his own license, called the GNU Public License. This new approach and ideology surrounding software creation took hold and eventually led to the formation of the Open Source Initiative in 1998.

Ubuntu

Canonical started sending out free compact discs with Ubuntu Linux in 2004 and quickly became popular for home users (many switching from Microsoft Windows). Canonical wants Ubuntu to be an easy to use graphical Linux desktop without need to ever see a command line. Of course they also want to make a profit by selling support for Ubuntu.

Debian

There is no company behind Debian. Instead there are thousands of well organised developers that elect a Debian Project Leader every two years. Debian is seen as one of the most stable Linux distributions. It is also the basis of every release of Ubuntu. Debian comes in three versions: stable, testing and unstable. Every Debian release is named after a character in the movie Toy Story.

Other

Distributions like Cent OS, Oracle Enterprise Linux and Scientific Linux are based on Red Hat Enterprise Linux and share many of the same principles, directories and system administration techniques. Linux Mint, Edubuntu and many other ubuntu named distributions are based on Ubuntu and thus share a lot with Debian. There are hundreds of other Linux distributions.

What is the Open Source Initiative?

The Open Source Initiative (OSI) was created to promote and protect open source software and communities.ii In short, the OSI acts as a central informational and governing repository of open source software. It provides rules and guidelines for how to use and interact with OSS, as well as providing code licensing information, support, definitions, and general community collaboration to help make the use and treatment of open source understandable and ethical.

How does OSS work?

Open source code is usually stored in a public repository and shared publicly. Anyone can access the repository to use the code independently or contribute improvements to the design and functionality of the overall project.

OSS usually comes with a distribution license. This license includes terms that define how developers can use, study, modify, and most importantly, distribute the software.iii According to the Synopsys Black Duck® KnowledgeBase, five of the most popular licenses are:

- MIT License
- GNU General Public License (GPL) 2.0-this is more restrictive and requires that copies of modified code are made available for public use
- Apache License 2.0
- GNU General Public License (GPL) 3.0
- BSD License 2.0 (3-clause, New or Revised)-this is less restrictive

When source code is changed, OSS must include what was altered as well as the methods involved. Depending on the license terms, the software resulting from these modifications may or may not be required to be made available for free.

What are some examples of OSS?

- GNU/Linux
- Mozilla Firefox
- VLC media player
- SugarCRM
- GIMP
- VNC
- Apache web server
- LibreOffice
- jQuery

COPA - Install Ubuntu Linux operating system and execute basic Linux commands

Handling commands and various editors

Objectives: At the end of this lesson you shall be able to

- know about terminal
- explain the command shell
- list out the directory layout of linux
- define the linux commands
- list out the special characters of linux OS
- explain various editors in linux OS.

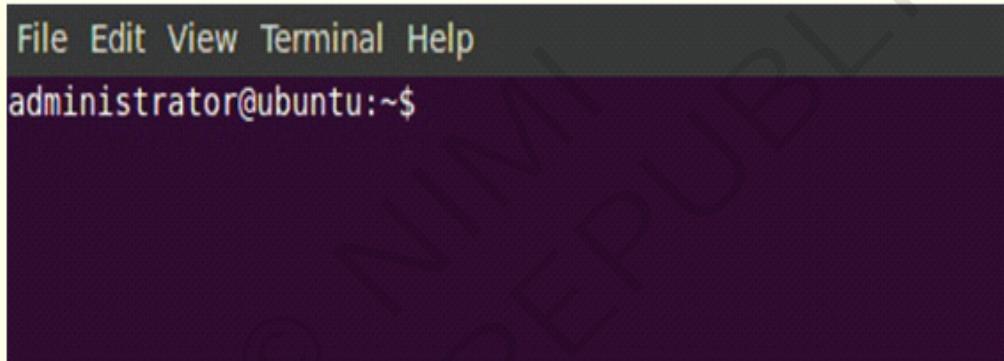
Starting up a terminal

To access the shell we will use a shell-like application, also called a terminal emulator. There is a huge number of good terminal applications out there, including the default ones in GNOME or KDE, or Yakuake, Guake, rxvt and so on. For now let's just stick with the default that

Some of the most popular shells are:

- **bash** - the Bourne-Again Shell, the default shell on most Linux systems.
- **sh** - the Bourne Shell, an older shell which is not so widely used anymore.

Fig 1



comes with your system. If you're using GNOME you can access the terminal by going to **Applications -> Accessories -> Terminal** or pressing Alt+F2 and typing gnome-terminal in the run box that appears, followed by Enter. If you're running KDE you can type instead **console** after pressing Alt+F2.

Depending on the distribution, the prompt may look something like **user@host\$**. The first part before the ampersand is the login username, and the other one is the hostname of the computer.

Command shell

A shell is a **command interpreter** which allows you to interact with the computer. The way things work is pretty simple: you type in commands, the shell interprets them, performs the tasks it was asked to do, and finally it sends the results to the standard output, which is usually the screen.

This is a list of files inside the root directory. The root directory is the first location in the file system tree hierarchy, and it is represented by the **slash** character **/**.

- **csh** - the 'C' Shell, which accepts a syntax which resembles the 'C' programming language.
- **tcsh** - an improved version of the 'C' Shell.
- **ksh** - the Korn Shell, initially developed in the early 1980's.
- **dash** - Debian Almquist Shell, a shell created by the Debian distribution.

Listing of shells available in the system

```
$ cat /etc/shells/
```

The above command will display the following output as on Fig 2.

In this tutorial we will focus on **Bash**, since it is the most widely used and also one of the most powerful shells out there. Bash is a modern implementation of the older Bourne Shell (**sh**), developed by the GNU project, which provides a huge amount of tools and which, together with the Linux kernel, desktop environments like GNOME or KDE and applications which run on top of them, comprise the whole Linux platform. On a Debian or Ubuntu distribution, the default shell used by the system is specified in the file **/etc/passwd** (default being Bash).

Fig 2

```
File Edit View Terminal Help
administrator@ubuntu:~$ cat /etc/shells
# /etc/shells: valid login shells
/bin/csh
/bin/sh
/usr/bin/es
/usr/bin/ksh
/bin/ksh
/usr/bin/rc
/usr/bin/tcsh
/bin/tcsh
/usr/bin/esh
/bin/dash
/bin/bash
/bin/rbash
/usr/bin/screen
administrator@ubuntu:~$
```

How to display default shell in the system

Type the following command in the terminal

\$ echo \$SHELL

And press Enter key. The default shell will be displayed as on Fig 3.

Fig 3

```
File Edit View Terminal Help
administrator@ubuntu:~$ echo $SHELL
/bin/bash
administrator@ubuntu:~$
```

The Linux directory layout

Directory	Description
	The nameless base of the file system. All other directories, files, drives, and devices are attached to this root. Commonly (but incorrectly) referred to as the "slash" or "/" directory. The "/" is just a directory separator, not a directory itself.
/bin	Essential command binaries (programs) are stored here (bash, ls, mount, tar, etc.)
/boot	Static files of the boot loader
/dev	Device files. In Linux, hardware devices are accessed just like other files, and they are kept under this directory.
/etc	Host-specific system configuration files.
/home	Location of users' personal home directories (e.g. /home/Susan).
/lib	Essential shared libraries and kernel modules.
/proc	Process information pseudo-file system. An interface to kernel data structures
/root	The root (super user) home directory.
/sbin	Essential system binaries (fdisk, fsck, init, etc.).
/tmp	Temporary files. All users have permission to place temporary files here.
/usr	The base directory for most shareable, read-only data (programs, libraries, documentation, and much more).
/usr/bin	Most user programs are kept here (cc, find, du, etc.).
/usr/include	Header files for compiling C programs.
/usr/lib	Libraries for most binary programs
/usr/local	"Locally" installed files. This directory only really matters in environments where files are stored on the network. Locally-installed files go in /usr/local/bin, /usr/local/lib, etc.). Also often used for Software packages installed from source, or software not officially shipped with the distribution.
/usr/sbin	Non-vital system binaries (lpd, useradd, etc.)
/usr/share	Architecture-independent data (icons, backgrounds, documentation, terminfo, man pages, etc.).
/usr/src	Program source code. E.g. The Linux Kernel, source RPMs, etc.
/usr/X11R6	The X Window System
/var	Variable data: mail and printer spools, log files, lock files, etc.

What are Linux commands?

Linux commands are executable binary files which can be ran to perform certain tasks, like for example listing the files in a directory running an entire graphical application. Examples of frequently used commands are ls, cd, pwd, date or cat. With the exception of executable files, there is also a category called shell built-ins, which are commands provided by the shell itself (Bash in our case). We'll deal with those later.

The general form of a Linux command is:

command options(s) filename(s)

Which specifies a command, followed by one or more parameters, and optionally one or more files to apply it on. For example:

```
$ echo -e 'Hello, world!\n'
```

Will output the text 'Hello, world!' followed by a newline character. The **-e** parameter (also called argument, or switch in this case) tells the echo command to interpret escaped characters, like the trailing **\n**, which will add a newline after the text inside the single quotes. Ignore the leading dollar sign, it just signifies the shell prompt.

A command may or may not have arguments. An argument can be an option or a filename.

Special characters in linux operating system

It is important to know that there are many symbols and characters that the shell interprets in special ways. This means that certain typed characters: a) cannot be used in certain situations, b) may be used to perform special operations, or, c) must be "escaped" if you want to use them in a normal way.

Character	Description
\	Escape character. If you want to reference a special character, you must "escape" it with a backslash first. Example: touch /tmp/filename*
/	Directory separator, used to separate a string of directory names. Example: /usr/src/linux
.	Current directory. Can also "hide" files when it is the first character in a filename.
..	Parent directory
~	User's home directory
*	Represents 0 or more characters in a filename, or by itself, all files in a directory. Example: pic*2002 can represent the files pic2002, picJanuary2002, picFeb292002, etc.
?	Represents a single character in a filename. Example: hello?.txt can represent hello1.txt, helloz.txt, but not hello22.txt
[]	Can be used to represent a range of values, e.g. [0-9], [A-Z], etc. Example: hello[0-2].txt represents the names hello0.txt, hello1.txt, and hello2.txt
	"Pipe". Redirect the output of one command into another command. Example: ls more
>	Redirect output of a command into a new file. If the file already exists, over-write it. Example: ls > myfiles.txt
>>	Redirect the output of a command onto the end of an existing file. Example: echo .Mary 555-1234. >> phonenumbers.txt
<	Redirect a file as input to a program. Example: more < phonenumbers.txt
;	Command separator. Allows you to execute multiple commands on a single line. Example: cd /var/log ; less messages

The cd command

The cd command is used to change the current directory (i.e., the directory in which the user is currently working) in Linux and other Unix-like operating systems. It is similar to the CD and CHDIR commands in MS-DOS.

cd's syntax is

cd [option] [directory]

The items in square brackets are optional. When used without specifying any directory name, cd returns the user to the previous current directory. This provides a convenient means of toggling between two directories.

When a directory name is provided, cd changes the current directory to it. The name can be expressed as an absolute pathname (i.e., location relative to the root directory) or as a local pathname (i.e., location relative to the current directory). It is usually more convenient to use a local pathname when changing to a subdirectory of the current directory.

As an example, the following would change the current directory, regardless of where it is on the system (because it is an absolute path), to the root directory (which is represented by a forward slash):

cd /

Likewise, the following would change the current directory, regardless of its location, to the /usr/sbin directory (which contains non-vital system utilities that are used by the system administrator):

cd /usr/sbin

If a user currently in the directory /usr/local/share/man/ desired to change to the directory /usr/local/share/man/man2, which is a subdirectory of the current directory, it would be possible to change by using the absolute pathname, i.e.,

cd /usr/local/share/man/man2

However, it would clearly be much less tedious to use the relative pathname, i.e.,

cd man2

On Unix-like operating systems the current directory is represented by a singledot and its parent directory (i.e., the directory that contains it) is represented by two consecutive dots. Thus, it is possible (and often convenient) to change to the parent of the current directory by using the following:

cd ..

Another convenient feature of cd is the ability for any user to return directly to its home directory by merely using a tilde as the argument. A home directory, also called a login directory, is the directory on a Unix-like operating system that serves as the repository for a user's personal files, directories and programs. It is also the directory that a user is first in after logging into the system. A tilde is a short, wavy, horizontal line character that represents the home directory of the current user. That is, any user can

return immediately to its home directory by typing the following and then pressing the Enter key:

cd ~

This is easier than typing the full name of the user's home directory, for instance, /home/josephine in the case of a user named josephine. (And it is just one of the numerous shortcuts that help make the command line on Unix-like operating systems so easy to use.)

When followed by a space and then a hyphen, cd both returns the user to the previous current directory and reports on a new line the absolute pathname of that directory. This can further enhance the already convenient toggling capability of cd. Toggling is particularly convenient when at least one of the two directories has a long absolute pathname, such as /usr/local/share/man/man2.

cd has only two options, and neither of them are commonly used. The -P option instructs cd to use the physical directory structure instead of following symbolic links. The -L option forces symbolic links to be followed.

The pwd command

The pwd command reports the full path to the current directory.

The current directory is the directory in which a user is currently operating while using a command line interface. A command line interface is an all-text display mode and it is provided via a console (i.e., a display mode in which the entire screen is text only) or via a terminal window (i.e., a text-only window in a GUI).

The full path, also called an absolute path, to a directory or file is the complete hierarchy of directories from the root directory to and including that directory or file. The root directory, which is designated by a forward slash (/), is the base directory on the filesystem (i.e., hierarchy of directories), and it contains all other directories, subdirectories and files on the system. Thus, the full path for any directory or file always begins with a forward slash.

pwd is one of the most basic commands in Linux and other Unix-like operating systems, along with ls, which is used to list the contents of the current directory, and cd, which is used to change the current directory.

pwd's syntax is

pwd [option]

Unlike most commands, pwd is almost always used just by itself, i.e.,

Pwd

That is, it is rarely used with its options and never used with arguments (i.e., file names or other information provided as inputs). Anything that is typed on the same line after pwd, with the exception of an option, is ignored, and no error messages are returned.

As an example, if a user with the username janis is in its home directory, then the above command would typically return /home/janis/ (because, by default, all home

directories are located in the directory /home). Likewise, if a user were currently working in directory /usr/share/config (which contains a number of program configuration files), then the same command would return /usr/share/config.

pwd is useful for confirming that the current directory has actually been changed to what the user intended after using cd. For example, after issuing the cd command to change the current directory from /home/janis to /usr/share/config, pwd could be used for confirmation; that is, the following sequence of commands would be issued:

```
cd /usr/share/config/  
pwd
```

The standard version of pwd has a mere two options, both of which are employed only infrequently. The --help option is used as follows:

```
pwd --help
```

This option displays information about pwd, of which there is very little because it is such a simple command (i.e., it only has two options and accepts no arguments).

The other option is --version, which displays the version number, i.e.,

```
pwd --version
```

Although it is often thought of as standing for present working directory, pwd is actually an acronym for print working directory. The word print is traditional UNIX terminology for write or display, and it originated when computer output was typically printed on paper by default because CRT (cathode ray tube) display monitors were not yet widely available.

The echo command

echo is a built-in command in the bash and C shells that writes its arguments to standard output.

A shell is a program that provides the command line (i.e., the all-text display user interface) on Linux and other Unix-like operating systems. It also executes (i.e., runs) commands that are typed into it and displays the results. bash is the default shell on Linux.

A command is an instruction telling a computer to do something. An argument is input data for a command. Standard output is the display screen by default, but it can be redirected to a file, printer, etc.

The syntax for echo is

```
$ echo $USER  
$ echo "Hello world"
```

The items in square brackets are optional. A string is any finite sequence of characters (i.e., letters, numerals, symbols and punctuation marks).

When used without any options or strings, echo returns a blank line on the display screen followed by the command prompt on the subsequent line. This is because pressing the ENTER key is a signal to the system to start a new

line, and thus echo repeats this signal.

When one or more strings are provided as arguments, echo by default repeats those strings on the screen. Thus, for example, typing in the following and pressing the ENTER key would cause echo to repeat the phrase This is a pen. on the screen:

```
echo This is a pen.
```

It is not necessary to surround the strings with quotes, as it does not affect what is written on the screen. If quotes (either single or double) are used, they are not repeated on the screen.

Fortunately, echo can do more than merely repeat verbatim what follows it. That is, it can also show the value of a particular variable if the name of the variable is preceded directly (i.e., with no intervening spaces) by the dollar character (\$), which tells the shell to substitute the value of the variable for its name.

For example, a variable named x can be created and its value set to 5 with the following command:

```
x = 5
```

The value of x can subsequently be recalled by the following:

```
echo The number is $x.
```

Echo is particularly useful for showing the values of environmental variables, which tell the shell how to behave as a user works at the command line or in scripts (short programs).

For example, to see the value of HOME, the environmental value that shows the current user's home directory, the following would be used:

```
echo $HOME
```

Likewise, echo can be used to show a user's PATH environmental variable, which contains a colon-separated list of the directories that the system searches to find the executable program corresponding to a command issued by the user:

```
echo $PATH
```

echo, by default, follows any output with a newline character. This is a non-printing (i.e., invisible) character that represents the end of one line of text and the start of the next. It is represented by \n in Unix-like operating systems. The result is that the subsequent command prompt begins on a new line rather than on the same line as the output returned by echo.

The -e option is used to enable echo's interpretation of additional instances of the newline character as well as the interpretation of other special characters, such as a horizontal tab, which is represented by \t. Thus, for example, the following would produce a formatted output:

```
echo -e "\n Projects: \n\ntplan \n\tcode \n\ttest\n"
```

(The above command should be written on a single line, although it may render as two lines on smaller display screens.) The -n option can be used to stop echo from adding the newline to output.

By making use of output redirection, echo provides a very simple way of creating a new file that contains text. This is accomplished by typing echo followed by the desired text, the output redirection operator (which is a rightward pointing angle bracket) and finally the name of the new file. The file can likewise be formatted by using special characters. Thus, for example, the formatted output from the above example could be used to create a new file called project1:

```
echo -e "\n Project1: \n\n\tplan \n\twrite \n\ttest\n" > project1
```

The contents of the new file, including any formatting, can be verified by using a command such as cat or less, i.e.,

```
less project1
```

echo can likewise be a convenient way of appending text to the end of a file by using it together with the append operator, which is represented by two consecutive rightward pointing angle brackets. However, there is always the risk of accidentally using a single bracket instead of two, thereby overwriting all of the contents of the file, and thus, this feature is best reserved for use in scripts.

echo can also be used with pattern matching, such as the wildcard character, which is represented by the star character. For example, the following would return the phrase The gif files are followed by the names of all the .gif image files in the current directory:

```
echo -e The gif files are *.gif
```

The cal command

Displays calendar of current month.

```
$ cal
```

July 2012

Su	Mo	Tu	We	Th	Fr	Sa
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

'cal' will display calendar for specified month and year.

```
$ cal 08 1991
```

August 1991

Su	Mo	Tu	We	Th	Fr	Sa
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

Date command

Display current time and date.

```
$ date
```

```
Fri Jul 6 01:07:09 IST 2012
```

If you are interested only in time, you can use 'date +%T' (in hh:mm:ss):

```
$ date +%T
```

```
01:13:14
```

tty command

Displays current terminal.

```
$ tty
```

```
/dev/pts/0
```

whoami command

This command reveals the current logged in user.

```
$ whoami
```

```
raghu
```

id command

This command prints user and groups (UID and GID) of current user.

```
$ id
```

```
uid=1000(raghu) gid=1000(raghu)
```

groups = 1000 (raghu), 4(adm), 20(dialout), 24(cdrom), 46(plugdev), 112(lpadmin), 120(admin), 122(sambashare)

By default information about current user is displayed. If another username is provided as an argument, information about that user will be printed:

```
$ id root
```

```
uid=0(root) gid=0(root) groups=0(root)
```

Clear command

This command clears the screen.

Getting help command

For all its advantages, a big disadvantage of command line is that there are a lot of commands and even more are their options and usage. But nobody can remember all commands. There are some smarter ways of using command line. Linux provides us with several such resources discussed here:

--help option

With almost every command, '--help' option shows usage summary for that command.

```
$ date --help
```

```
Usage: date [OPTION]... [+FORMAT]
```

or: date [-u|--utc|--universal] [MMDDhhmm[[CC]YY].[ss]]

Display the current time in the given FORMAT, or set the system date.

The whatis command

The whatis command provides very brief descriptions of command line programs (i.e., all-text mode programs) and other topics related to Linux and other Unix-like operating systems.

It accomplishes this by searching the short descriptions in the whatis database for each keyword provided to it as an argument (i.e., input data). This database contains just the title, section number and description from the NAME section of each page in the man manual that is built into most Unix-like systems.

The syntax for whatis is:

whatis keyword(s)

For example, the following provides a single line summary of the head command (which by default displays the first ten lines of each file that is provided to it as an argument):

whatis head

whatis can be used to simultaneously search for information about multiple topics. For example, the following would provide information about both head and tail (which by default reads the final ten lines of files):

whatis head tail

The output of whatis is limited by the fact that it provides only a single line for each keyword found in the database; thus it supplies incomplete information about even moderately complex commands. For example, the following use of whatis to obtain information about the cat command generates the output "concatenate files and print on the standard output":

whatis cat

However, this omits some important information about cat, particularly the facts that it is very convenient to use for reading files and that it is also used to create and write to files.

whatis is similar to the apropos command. However, apropos is more powerful in that its arguments are not limited to complete words but can also be strings (i.e., any finite sequences of characters) which comprise parts of words. Both commands are unusual in that they have no options.

The man command (which is used to read the built-in manual pages), when used with its -f option, produces the same output as whatis. Thus, for example,

man -f cat

is equivalent to

whatis cat

Info pages

Info documents are sometimes more elaborated than man pages. But for some commands, info pages are just the same as man pages. These are like web pages. Internal links are present within the info pages. These links are called nodes. Info pages can be navigated from one page to another through these nodes.

\$ info date

Word processors in the Linux environment

Text editors are used by many different types of people. Casual users, writers, programmers, and system administrators will all use a text editor at one time or another in Linux.

Use of text editor

A text editor is just like a word processor without a lot of features. All operating systems come with a basic text editor. Linux comes with several. The main use of a text editor is for writing something in plain text with no formatting so that another program can read it. Based on the information it gets from that file, the program will run one way or another.

Movement commands:

Depending on your system, the arrow keys or the backspace key may not work. Instead, you can use these commands to perform the same tasks.

To	Hold down Ctrl key and press	Instead of
Delete a character	backspace	backspace
Move up a line	p	up arrow
Move down a line	n	down arrow
Move left one space	b	left arrow
Move right one space	f	right arrow
Move to the end of line	e	end

vi	typed at the command line to open one or more files in the same directory (vi tomato.txt opens a file named "tomato.txt" in the current directory) (vi parsley sage rosemary opens the three files "parsley," "sage" and "rosemary" in the current directory)
vi *	typed at the command line to open every file in the current directory
:q	closes (quits) a file to which no changes have been made
:q!	quits without saving any changes
:w	writes (i.e., saves) the current file to disk
:wq	writes the buffer contents to disk (i.e., saves changes) and quits
ZZ	same as :wq
i	activates text insert mode, inserting text immediately under the current position of the cursor.
h	moves the cursor one character to the left (2h moves the cursor two characters to the left)
j	moves the cursor one line down (3j moves the cursor three lines down)
k	moves the cursor one line up
l	moves the cursor one character to the right
G	moves the cursor to the desired line; moves the cursor to the last line of text if not preceded by a modifying integer (5G moves the cursor to the fifth line)
a	switches to insert mode and allows insertion of text immediately to the right of the cursor.
x	deletes the character immediately under the cursor (xxx deletes the character immediately under cursor and then deletes the two characters to its right)
X	deletes a single character to the left of cursor
D	removes the text on the current line from the character under the cursor to the end of the line
dw	deletes the character immediately under the cursor and the remaining characters to the right of it in the same word (2dw deletes the character immediately under the cursor, the remaining characters to the right of it in same word and all of the next word)
dd	deletes the entire line containing the cursor, and the cursor then moves to the next line (2dd deletes two consecutive lines beginning with the current line)
cw	deletes the character under the cursor and to its right in the same word and allows new characters to be typed in to replace them (2cw deletes the character under the cursor and to its right in the same word and in the next word, and then allows replacement characters to be typed in)

cc	erases the current line and allows replacement text to be typed in (2cc erases the current line and the next line and allows replacement text to be typed in for both lines)
cb	deletes the characters to the left of the cursor in the current word and allows replacement characters to be typed in (3cb deletes the characters to the left of the cursor in the current word together with the two words to its left and then allows replacement text to be typed in)
R	activates text input mode allowing text under and to the right of the cursor to be overwritten one character at a time
xp	transposes two adjacent characters
deep	transposes two adjacent words
ddp	transposes two adjacent lines
~	changes case of the character under the cursor
J	joins the current line with the next line
u	reverses the effects of the most recent command that has changed the buffer
U	undoes all changes made to the current line during the current visit to it
:s/	searches the text for the first instance of a designated string (:s/cucumber searches the text for the first instance of the string "cucumber")
n	searches the text for the next instance of a designated string
:s// /	replaces the first instance of a designated string (:s/cucumber/radish/ replaces the first instance of the string "cucumber" with the string "radish")
:%s//	replaces every instance of a designated string (:%s/cucumber/radish/ replaces every instance of the string "cucumber" with the string "radish")
:r	inserts text into the currently open file from another file (:r lettuce.txt inserts text into the currently open file from the file named "lettuce.txt")
:w>>	appends the text from the currently open file into another file (:w>> cabbage appends the text from the currently open file into the file named "cabbage")

Managing files and directories

Objectives: At the end of this lesson you shall be able to

- define manipulating files and directories
 - define basic file commands
 - explain other file commands
 - define additional useful commands in linux OS.
-

Manipulating files or directories: Using Linux isn't different from any other computer operating system. You create, delete, and move files on

your hard drive in order to organize your information and manage how your system works or looks. This section shows you how to do these tasks quickly and easily.

Although the graphical interface for Linux, the X Window System, may offer drag and drop or multiple selections in order to copy or delete files, many of the commands you'll learn here form the base of these operations. It is worth knowing how these programs work, even if you don't use Linux in the console mode.

Working with files: In this chapter we learn how to recognise, create, remove, copy and move files using commands like file, touch, rm, cp, mv and rename, etc...

All files are case sensitive: Files on Linux (or any Unix) are case sensitive. This means that FILE1 is different from file1, and /etc/hosts is different from /etc/Hosts (the latter one does not exist on a typical Linux computer).

The file command: The file command attempts to classify each filesystem object (i.e., file, directory or link) that is provided to it as an argument (i.e., input). Thus, it can usually provide immediate information as to whether some specified object is, for example, a GIF89a image file, a directory, a GNU tar archive, ASCII English text, a symbolic link, an HTML document, an empty file, bzip2 compressed data, an ELF 32-bit LSB executable, etc.

File accomplishes this by probing each object with three types of tests until one succeeds. The first is a filesystem test, which uses the stat system call to obtain information from the object's inode (which contains information about a file). A system call is a request in a Unix-like operating system for a service performed by the kernel (i.e., the core of the operating system).

The second test checks to see if there is a magic number, which is a number embedded at or near the beginning of many types of files that indicates the file format(i.e., the type of file).

In the event that the first two tests fail to determine the type of a file, language tests are employed to determine if it is plain text (i.e., composed entirely of human-readable characters), and, if so, what type of plain text, such as HTML (hypertext markup language) or source code (i.e., the original version of a program as written by a human). In this situation, file also attempts to determine the natural language (e.g., English, Turkish or Japanese) that is used in the file.

A simplified version of file's syntax is

file [option(s)] object_name(s)

File has several options, but it is most commonly used without any of them. For example, information about a file named file1 that is located in the in the current directory (i.e., the directory in which the user is currently working) could be obtained by merely typing the following and pressing the RETURN key:

file file1

Information about the types of all of the files in the current directory can be obtained by using the star wildcard to represent every object in that directory as follows:

file *

Likewise, information about all of the files in another directory can be obtained by using that directory as an argument and following it immediately by a forward slash and the star wildcard. For example, the following classifies all of the objects in the /boot directory:

file /boot/*

The square brackets wildcard can be used together with the star wildcard to show the file types for only those objects whose names begin with specified letters or with a specified range of letters. For example, the following would show only those objects in the current directory whose names begin with letters a through g:

file [a-g]*

The -k option tells file to not stop at the first successful test, but to keep going; this can result in the reporting of additional information about some filesystem objects. The -b (i.e., brief) option tells file to not prepend filenames to output lines, which can be useful when compiling statistics about file types. The -v option returns information about the version of file that is installed.

Creating files and directories command

mkdir command

The mkdir command is used to create new directories.

A directory, referred to as a folder in some operating systems, appears to the user as a container for other directories and files. However, Unix-like operating systems treat directories as merely a special type of file that contains a list of file names and their corresponding inode numbers. Each inode number refers to an inode, which is located in inode tables (which are kept at strategic locations around the filesystem) and which contains all information about a file (e.g., size, permissions and date of creation) except its name and the actual data that the file contains.

mkdir has the following example

\$ mkdir example

\$ ls -l

total 4

drwxr-xr-x 2 raghu raghu 4096 2012-07-06 14:09 example

directory_name is the name of any directory that the user is asking mkdir to create. Any number of directories can be created simultaneously.

Thus, for example, the following command would create three directories within the current directory (i.e., the directory in which the user is currently working) with the names dir_1, dir_2 and dir_3:

mkdir dir_1 dir_2 dir_3

If a directory name provided as an argument (i.e., input) to mkdir is the same as that of an existing directory or file in the same directory in which the user is asking mkdir to create the new directory, mkdir will return a warning message such as mkdir: cannot create directory `dir_1':

File exists and will not create a file with that name. However, it will then continue to create directories for any other names provided as arguments.

It is necessary for a user to have write permission (i.e., permission from the system to create or change a file or directory) in the parent directory (i.e., the directory in which the new directory is to be created) in order to be able to create a new directory.

Directories created by mkdir automatically include two hidden directories, one representing the directory just created (and represented by a single dot) and the other representing its parent directory (and represented by two consecutive dots). This can be seen by using the ls (i.e., list) command with its -a option, which tells ls to show all directories and files, (including hidden ones) in any directory provided to it as an argument, or in the current directory if there are no arguments, i.e.,

```
ls -a
```

mkdir's -m option is used to control the permissions of new directories. New directories are by default created with the read, write and execute (i.e., run as a program if a program) permissions enabled for the owner (i.e., the creator of the directory by default) and group and the read and execute permissions enabled for other users. Thus, for example, to create a directory named dir_4 for which all three types of permissions were enabled for all users, the sequence 777 would be employed after -m, for example:

```
mkdir -m 777 dir_4
```

The first digit represents the owner, the second represents the group and the third represents other users. The number 7 represents all three types of permission (i.e., read, write and execute), 6 stands for read and write only, 5 stands for read and execute, 4 is read only, 3 is write and execute, 2 is write only, 1 is execute only and 0 is no permissions.

Thus, for example, to create a new directory named dir_5 for which the owner has read and write permissions, the group has read permission and other users have no permissions, the following would be used:

```
mkdir -m 640 dir_5
```

The -p (i.e., parents) option creates the specified intermediate directories for a new directory if they do not already exist. For example, it can be used to create the following directory structure:

```
mkdir -p food/fruit/citrus/oranges
```

It is very easy to confirm that this series of directories has been created by using the du (i.e., disk usage) command with the name of the first directory as an argument. In the case of the above example this would be

du food

Other options include -v (i.e., verbose), which returns a message for each created directory, --help, which returns brief information about mkdir, and --version, which returns the version number of the currently installed mkdir program

Touch command: The touch command updates the access and modification times of each FILE to the current system time.

If you specify a FILE that does not already exist, touch creates an empty file with that name.

If the FILE argument is a dash ("") is handled specially and causes touch to change the times of the file associated with standard output.

```
$ touch file1 file2 file3
```

```
$ ls -l
```

```
total 4
```

```
drwxr-xr-x 2 raghu raghu 4096 2012-07-06 14:09 example  
-rw-r--r-- 1 raghu raghu 0 2012-07-06 14:20 file1  
-rw-r--r-- 1 raghu raghu 0 2012-07-06 14:20 file2  
-rw-r--r-- 1 raghu raghu 0 2012-07-06 14:20 file3
```

A feature of touch is that, in contrast to some commands such as cp (which is used to copy files and directories) and mv (which is used to move or rename files and directories), it does not automatically overwrite (i.e., erase the contents of) existing files with the same name. Rather, it merely changes the last access times for such files to the current time.

Several of touch's options are specifically designed to allow the user to change the timestamps for files. For example, the -a option changes only the access time, while the -m option changes only the modification time. The use of both of these options together changes both the access and modification times to the current time, for example:

```
touch -am file3
```

The -r (i.e., reference) option followed directly by a space and then by a file name tells touch to use that file's time stamps instead of current time. For example, the following would tell it to use the times of file4 for file5:

```
touch -r file4 file5
```

The -B option modifies the timestamps by going back the specified number of seconds, and the -F option modifies the time by going forward the specified number of seconds. For example, the following command would make file7 30 seconds older than file6.

```
touch -r file6 -B 30 file7
```

The -d and -t options allow the user to add a specific last access time. The former is followed by a string (i.e., sequence of characters) in the date, month, year, minute:second format, and the latter uses a [[CC]YY]MMDDhhmm[.ss] format. For example, to change the last access time of file8 to 10:22 a.m. May 1, 2005, 1 May 2005 10:22 would be enclosed in single quotes and used as follows, i.e.:

```
touch -d '1 May 2005 10:22' file8
```

Partial date-time strings can be used. For example, only the date need be provided, as shown for file9 below (in which case the time is automatically set to 0:00):

```
touch -d '14 May' file9
```

Just providing the time, as shown below, automatically changes the date to the current date:

touch -d '14:24' file9

The most commonly used way to view the last modification date for files is to use the ls command with its -l option. For example, in the case of a file named file10 this would be

ls -l file10

The complete timestamps for any file or directory can be viewed by using the stat command. For example, the following would show the timestamps for a file named file11:

stat file11

The --help option displays a basic list of options, and the --version option returns the version of the currently installed touch program.

Copy, move and remove commands

Copy command: The cp command is used to copy files and directories. The copies become independent of the originals (i.e., a subsequent change in one will not affect the other).

cp's basic syntax is

\$cp source destination

As a safety precaution, by default cp only copies files and not directories. If a file with the same name as that assigned to the copy of a file (or a directory with the same name as that assigned to the copy of a directory) already exists, it will be overwritten (i.e., its contents will be lost). However, the owner, group and permissions for the copy become the same as those of the file with the same name that it replaced. The last access time of the source file and the last modification time of the new file are set to the time the copying was performed.

When a copy is made of a file or directory, the copy must have a different name than the original if it is to be placed in the same directory as the original. However, the copy can have the same name if it is made in a different directory. Thus, for example, a file in the current directory (i.e., the directory in which the user is currently working) named file1 could be copied with the same name into another directory, such as into /home/john/, as follows:

cp file1 /home/john/file1

Any number of files can be simultaneously copied into another directory by listing their names followed by the name of the directory. cp is an intelligent command and knows to do this when only the final argument (i.e., piece of input data) is a directory. The files copied into the directory will all have the same names as the originals. Thus, for example, the following would copy the files named file2, file3 and file4 into a directory named dir1:

cp file2 file3 file4 dir1

The -r (i.e., recursive) option, which can also be written with an upper case R, allows directories including all of their contents to be copied. (Directories are not copied by default in order to make it more difficult for users to accidentally overwrite existing directories which have the same name as that assigned to the copy being made and

which might contain critical directory structures or important data.) Thus, for example, the following command would make a copy of an existing directory called dir2, inclusive of all its contents (i.e., files, subdirectories, their subdirectories, etc.), called dir3:

cp -r dir2 dir3

The -i (i.e., interactive) option prompts the user in the event that any name assigned to a copy is already in use by another file and that file would thus be overwritten. Entering the letter y (either lower case or upper case) in response to the prompt causes the command to continue; any other answer prevents the command from overwriting the file. Thus, for example, if it is desired to make a copy of a directory called dir4 and call it dir5 and if a directory named dir4 already exists, the following would prompt the user prior to replacing any files with identical names in the latter directory:

cp -ri dir4 dir5

The -a option preserves as much of the structure and attributes of the original directory and its contents as possible in the new directory and is thus useful for creating archives. It is similar to the -r option in that it copies directories recursively; however, it also never follows symbolic links. It is equivalent to the -rdp combination of options.

All the files in a directory can be copied to another directory by using the star wildcard. The star character represents any single character or any combination of characters. Thus, for example, the following would copy all of the files in a directory named dir6 into another existing directory called dir7:

cp dir6/* dir7

cp can also be used with the star wildcard or other pattern matching characters to selectively copy files and directories. For example, to copy all of the files in the current directory that have the filename extension .html into another existing directory called dir8, the following would be used:

cp *.html dir8

In this case, the star wildcard represents anything whose name ends with the .html extension.

Among the other options for cp are -b, which makes backup copies of each destination file, -f (i.e., force), which removes destination files that cannot be opened and tries again, -s, which makes symbolic links instead of copying, -u (i.e., update), which copies only if the source file is newer than the destination file or if the destination file is missing, -v (i.e., verbose), which makes brief comments about what is going on, and -x, which tells cp to stay on the same filesystem.

Move command: The mv command is used to rename and move files and directories. Its general syntax is:

\$ mv source destination

The arguments are names of files and directories. If two file names are provided as arguments, mv renames the

first as the second. If a list of arguments is provided and the final argument in the sequence is the name of an existing directory, mv moves all of the other items into that directory. If the final argument is not an existing directory and more than two arguments are provided, an error message is returned.

If the destination file is located in the same directory as the source file, then the source file can only be renamed. If both are in different directories, then the source file is moved to the directory named in the destination argument, in which it can keep its original name or be assigned a new name. If the target is a directory, then the source file or directory is moved into that directory and retains its original name.

Thus, for example, the following would rename a file called file1 to file2, while keeping it in the current directory (i.e., the directory in which the user is currently working):

mv file1 file2

The following would move a file named file3, without changing its name, from the current directory to an existing subdirectory of the current directory named dir1:

mv file3 dir1/file3

mv can be used to move any number of files and directories simultaneously. For example, the following command moves all files and directories, including all the contents of those directories, from the current directory to the directory /home/alice/new/:

mv * /home/alice/new/

The asterisk is a wildcard character that represents any string (i.e., sequence of characters). Thus, in the above example it represents the name of every file and directory in the current directory.

mv makes it as easy to move a file or directory up the hierarchy of directories (i.e., closer to the root directory) as down it. For example, the following would move a file named file4, which is currently located in the sub-subdirectory dir/dir/ of the user's home directory, to the top level in the user's home directory:

mv dir/dir/file4 ~

The root directory is the directory that contains all other directories on a Unix-like operating system and which is at the top of the hierarchy of directories. A user's home directory is the directory in which a user finds itself by default after logging into the system and which can be represented by the tilde (wavy horizontal linecharacter).

By default, mv does not provide any confirmation on the display screen if its action is completed without problems. This is consistent with the rule of silence tenet of the Unix philosophy.

Thus it is wise for users new to Unix-like operating systems to always use the -i option, which makes mv interactive in the situation in which files and/or directories with the same name already exist in the destination directory. For example, the above command would be made interactive as follows:

mv -i * /home/alice/new/

Among mv's few other options are -b, which tells it to make a backup copy of each file that would otherwise be overwritten or removed, and -v, which tells it to be verbose and display the name of each file before moving it. Detailed information (including all options) about mv can be obtained by using its --help option, and information about the current version can be obtained by using its --version option.

Remove or Delete

\$ rmdir

'rmdir' command removes any empty directories, but cannot delete a directory if a file is present in it. To use 'rmdir' command, you must first remove all the files present the directory you wish to remove (and possibly directories if any).

Remove files and directories: The rm (i.e., remove) command is used to delete files and directories on Linux and other Unix-like operating systems.

The general syntax for rm is:

rm [options] [-r directories] filenames

The items in square brackets are optional. When used just with the names of one or more files, rm deletes all those files without requiring confirmation by the user. Thus, in the following example, rm would immediately delete the files named file1, file2 and file3, assuming that all three are located in the current directory (i.e., the directory in which the user is currently working):

rm file1 file2 file3

Error messages are returned if a file does not exist or if the user does not have the appropriate permission to delete it. Write-protected files prompt the user for a confirmation (with a y for yes and an n for no) before removal. Files located in write-protected directories can never be removed, even if those files are not write-protected.

The -f (i.e., force) option tells rm to remove all specified files, whether write-protected or not, without prompting the user. It does not display an error message or return error status if a specified file does not exist. However, if an attempt is made to remove files in a write-protected directory, this option will not suppress an error message.

The -i (i.e., interactive) option tells rm to prompt the user for confirmation before removing each file and directory. If both the -f and -i options are specified, the last one specified takes affect.

As a safety measure, rm does not delete directories by default. In order to delete directories, it is necessary to use the -r option, which is the same as the -R option. This option recursively removes directories and their contents in the argument list; that is, the specified directories will first be emptied of any subdirectories (including their subdirectories and files, etc.) and files and then removed. The user is normally prompted for removal of any write-protected files in the directories unless the -f option is used.

If a file encountered by rm is a symbolic link, the link is removed, but the file or directory to which that link refers will not be affected. A user does not need write permission to delete a symbolic link, as long as the user has write permission for the directory in which that link resides.

The rm command supports the -- (two consecutive dashes) parameter as a delimiter that indicates the end of the options. This is useful when the name of a file or directory begins with a dash or hyphen. For example, the following removes a directory named -dir1:

```
rm -r -- -dir1
```

Other options include -v (i.e., verbose), which provides additional information about what is happening, --help, which provides basic documentation about rm, and --version, which tells the version of rm that is currently in use. Some differences exist among the various versions of rm, so it is always wise to read the documentation for the particular system.

The rmdir command differs from rm in that it is only used to remove empty directories

The rmdir command

The rmdir command is used to remove empty directories in Linux

The syntax for rmdir is

```
rmdir [option] directory_names
```

When used without any options, rm will delete any empty directories whose names are supplied as arguments (i.e., inputs) regardless of whether such directories have write permission or not. Thus, for example, the following command would remove two empty directories named dir1 and dir2 that are located in the current directory (i.e., the directory in which the user is currently working):

```
rmdir dir1 dir2
```

The ability to remove only empty directories is a built-in safeguard that helps prevent the accidental loss of data. This is important because once deleted, it is extremely difficult or impossible to recover deleted data on Unix-like operating systems¹.

The -p (i.e., parents) option tells rmdir to remove the parent directories of the specified directory if each successive parent directory will, in turn, become empty and if each parent directory has write permission. Thus, for example, the following would remove dir5, dir4 and dir3 if dir5 were empty, dir4 only contained dir5 and dir3 only contained dir4 (which, in turn, contained dir5):

```
rmdir -p dir3/dir4/dir5
```

This provides a symmetry with the -p option of the mkdir command, which is used to create directories. Thus, the above set of nested directories could be easily created with the following:

```
mkdir -p dir3/dir4/dir5
```

In contrast to the rm command, which is used to delete both files and directories, there is no -r option for rmdir. at

least on the GNU version that is standard on Linux. That option allows rm to recursively delete a directory by first deleting all of its contents, beginning with those in the lowest levels of subdirectories. Thus, if a user wants to remove an entire directory structure, it is usually most efficient to use rm with its -r option rather than trying to first remove the contents of each directory, its subdirectories, etc.

Three options that rmdir shares with rm are -v (i.e., verbose), which provides additional information about what is happening, --help, which provides basic documentation about rmdir, and --version, which tells the version of rmdir that is currently in use. Some differences exist among the various versions of rmdir, so it is always wise to read the documentation for the particular system.

Listing and combining files with the cat command

The cat (concatenate file) command is used to send the contents of files to your screen. This

command may also be used to send files' contents into other files. Hour 6 covers terms such as standard input, standard output, and redirection, and this section shows you some basic uses for this command.

Although cat may be useful for reading short files, it is usually used to either combine, create, overwrite, or append files. To use cat to look at a short file, you can enter

```
$ cat test.txt
```

This text file was created by the cat command.

Cat could be the world's simplest text editor.

If you read this book, you'll learn how to use cat.

This is the last line of text in this file.

The cat command also has a number of options. If you'd like to see your file with line

numbers, perhaps to note a specific phrase, you can use the -n option:

```
$ cat -n test.txt
```

- This text file was created by the cat command.
- Cat could be the world's simplest text editor.
- If you read this book, you'll learn how to use cat.
- This is the last line of text in this file.

and also use cat to look at several files at once, because cat accepts wildcards, for example:

```
$ cat -n test*
```

- This text file was created by the cat command.
- Cat could be the world's simplest text editor.
- If you read this book, you'll learn how to use cat.
- This is the last line of text in this file.
- This is the first line of test2.txt.
- This file was also created by cat.
- This is the last line of test2.txt.

As you can see, cat has also included a second file in its output, and has numbered each line of the output, not each file. Note that also see both files with

\$ cat test.txt test2.txt

The output will be exactly the same as if had used a wildcard. But looking at several files is only one way to use cat. You can also use the cat command with the redirection operator > to combine files. For example, if you would like to combine test.txt and test2.txt into a third file called test3.txt, you can use

\$ cat test* > test3.txt

check the result with

ls -l test*

In this case, user first decide whether you want the contents of test.txt to go into test2.txt, or the contents of test2.txt to go into test.txt . Then, using cat with the >> redirection operator, you might type

\$ cat test.txt >> test2.txt

This appends the contents of test.txt to the end of the test2.txt . To check the results, use cat again:

\$ cat test2.txt

This is the first line of test2.txt.

This file was also created by cat.

This is the last line of test2.txt.

This text file was created by the cat command.

Cat could be the world's simplest text editor.

If you read this book, you'll learn how to use cat.

This is the last line of text in this file.

Note that if you had entered the command.

\$ cat -n test.txt >> test2.txt

The test2.txt file would look like

\$ cat test2.txt

This is the first line of test2.txt.

This file was also created by cat.

This is the last line of test2.txt.

- This text file was created by the cat command.
- Cat could be the world's simplest text editor.
- If you read this book, you'll learn how to use cat.
- This is the last line of text in this file.

Finally, here's a trick you can use if you want to create a short text file without running a word processor or text editor. Because the cat command can read the standard input (more about this in Hour 6), you can make the cat command create a file and fill it with your keystrokes.

Here's how:

\$ cat > myfile.txt

Now, enter some text:

\$ cat > myfile.txt

This is the cat word processor.

This is the end of the file.

Then, when you're done typing, press Ctrl+D to close the file. To see if this works, try

\$ ls -l myfile.txt

-rw-rw-r-- 1 bball bball 61 Nov 12 18:26 myfile.txt

\$ cat myfile.txt

This is the cat word processor.

This is the end of the file.

user should also know that the cat command will print out the contents of any file, and not

just text files. Although cat may be useful to look at one or several short files,

Date command examples to display and set system date time

Date command is helpful to display date in several formats. It also allows you to set systems date and time.

Here few examples on how to use date command with practical examples.

When execute date command without any option, it will display the current date and time as shown below.

\$ date

Mon May 20 22:02:24 PDT 2013

1 Display Date from a String Value using -date Option

If you have a static date or time value in a string, you can use -d or -date option to convert the input string into date format as shown below.

Please note that this doesn't use the current date and time value. Instead is uses the date and time value that you pass as string.

The following examples takes an input date only string, and displays the output in date format. If you don't specify time, it uses 00:00:00 for time.

\$ date --date="12/2/2014"

Tue Dec 2 00:00:00 PST 2014

\$ date --date="2 Feb 2014"

Sun Feb 2 00:00:00 PST 2014

\$ date --date="Feb 2 2014"

Sun Feb 2 00:00:00 PST 2014

The following example takes an input date and time string, and displays the output in date format.

\$ date --date="Feb 2 2014 13:12:10"

Sun Feb 2 13:12:10 PST 2014

2 Read Date Patterns from a file using -file option

This is similar to the -d or -date option that we discussed above. But, you can do it for multiple date strings. If you have a file that contains various static date strings, you can use -f or -file option as shown below.

In this example, we can see that datefile contained 2 date strings. Each line of datefile is parsed by date command and date is outputted for each line.

```
$ cat datefile
```

Sept 9 1986

Aug 23 1987

```
$ date --file=datefile
```

Tue Sep 9 00:00:00 PDT 1986

Sun Aug 23 00:00:00 PDT 1987

3 Get Relative Date Using -date option

You can also use date command to get a future date using relative values.

For example, the following examples gets date of next Monday.

```
$ date --date="next mon"
```

Mon May 27 00:00:00 PDT 2013

If string=@ is given to date command, then date command convert seconds since the epoch (1970-01-01 UTC) to a date.

It displays date in which 5 seconds are elapsed since epoch 1970-01-01 UTC:

```
$ date --date=@5
```

Wed Dec 31 16:00:05 PST 1969

It displays date in which 10 seconds are elapsed since epoch 1970-01-01 UTC:

```
$ date --date=@10
```

Wed Dec 31 16:00:10 PST 1969

It displays date in which 1 minute (i.e. 60 seconds) is elapsed since epoch 1970-01-01 UTC:

```
$ date --date=@60
```

Wed Dec 31 16:01:00 PST 1969

4 Display past date

You can display a past date using the -date command. Few possibilities are shown below.

```
$ date --date='3 seconds ago'
```

Mon May 20 21:59:20 PDT 2013

```
$ date --date="1 day ago"
```

Sun May 19 21:59:36 PDT 2013

```
$ date --date="yesterday"
```

Sun May 19 22:00:26 PDT 2013

```
$ date --date="1 month ago"
```

Sat Apr 20 21:59:58 PDT 2013

```
$ date --date="1 year ago"
```

Sun May 20 22:00:09 PDT 2012

5 Set Date and Time using -set option

You can set date and time of your system using -s or -set option as shown below..

In this example, initially it displayed the time as 20:09:31. We then used date command to change it to 21:00:00.

```
$ date
```

Sun May 20 20:09:31 PDT 2013

```
$ date -s "Sun May 20 21:00:00 PDT 2013"
```

Sun May 20 21:00:00 PDT 2013

```
$ date
```

Sun May 20 21:00:05 PDT 2013

6 Display Universal Time using -u option

You can display date in UTC format using -u, or -utc, or -universal option as shown below.

```
$ date
```

Mon May 20 22:07:53 PDT 2013

```
$ date -u
```

Tue May 21 05:07:55 UTC 2013

7 Display Last Modification Time using -r option

In this example, the current time is 20:25:48

```
$ date
```

Sun May 20 20:25:48 PDT 2013

The timestamp of datefile is changed using touch command. This was done few seconds after the above date command's output.

```
$ touch datefile
```

The current time after the above touch command is 20:26:12

```
$ date
```

Sun May 20 20:26:12 PDT 2013

Finally, use the date command -r option to display the last modified timestamp of a file as shown below. In this example, it displays last modified time of datefile as 20:25:57. It is somewhere between 20:25:48 and 20:26:12 (which is when we execute the above touch command to modify the timestamp).

```
$ date -r datefile
```

Sun May 20 20:25:57 PDT 2013

8 Various Date command formats

You can use formatting option to display date command in various formats using the following syntax:

```
$ date +%<format-option>
```

Command	Description
• apropos whatis	Show commands pertinent to string. See also threadsafe
• man -t ascii ps2pdf - > ascii.pdf	make a pdf of a manual page
which command	Show full path name of command
time command	See how long a command takes
• time cat	Start stopwatch. Ctrl-d to stop. See also sw
dir navigation	
• cd -	Go to previous directory
• cd	Go to \$HOME directory
(cd dir && command)	Go to dir, execute command and return to current dir
• pushd .	Put current dir on stack so you can popd back to it
file searching	
• alias l='ls -l --color=auto'	quick dir listing. See also l
• ls -lrt	List files by date. See also newest and find_mm_yyyy
• ls /usr/bin pr -T9 -W\$COLUMNS	Print in 9 columns to width of terminal
find -name '*.[ch]' xargs grep -E 'expr'	Search 'expr' in this dir and below. See also findrepo
find -type f -print0 xargs -r0 grep -F 'example'	Search all regular files for 'example' in this dir and below
find -maxdepth 1 -type f xargs grep -F 'example'	Search all regular files for 'example' in this dir
find -maxdepth 1 -type d while read dir; do echo \$dir; echo cmd2; done	Process each item with multiple commands (in while loop)
• find -type f ! -perm -444	Find files not readable by all (useful for web site)
• find -type d ! -perm -111	Find dirs not accessible by all (useful for web site)
• locate -r 'file[^/]*\.txt'	Search cached index for names. This re is like glob *file*.txt
• look reference	Quickly search (sorted) dictionary for prefix
• grep --color reference /usr/share/dict/words	Highlight occurrences of regular expression in dictionary
archives and compression	
gpg -c file	Encrypt file
gpg file.gpg	Decrypt file
tar -c dir/ bzip2 > dir.tar.bz2	Make compressed archive of dir/
bzip2 -dc dir.tar.bz2 tar -x	Extract archive (use gzip instead of bzip2 for tar.gz files)
tar -c dir/ gzip gpg -c ssh user@remote 'dd of=dir.tar.gz.gpg'	Make encrypted archive of dir/ on remote machine

	<code>find dir/ -name '*.txt' xargs cp -a --target-directory=dir_txt/ --parents</code>	Make copy of subset of dir/ and below
	<code>(tar -c /dir/to/copy) (cd /where/to/ && tar -x -p)</code>	Copy (with permissions) copy/ dir to /where/to/ dir
	<code>(cd /dir/to/copy && tar -c .) (cd /where/to/ && tar -x -p)</code>	Copy (with permissions) contents of copy/ dir to /where/to/
	<code>(tar -c /dir/to/copy) ssh -C user@remote 'cd /where/to/ && tar -x -p'</code>	Copy (with permissions) copy/ dir to remote:/where/to/ dir
	<code>dd bs=1M if=/dev/sda gzip ssh user@remote 'dd of=sda.gz'</code>	Backup harddisk to remote machine
rsync (Network efficient file copier: Use the --dry-run option for testing)		
	<code>rsync -P rsync://rsync.server.com/path/to/file file</code>	Only get diffs. Do multiple times for troublesome downloads
	<code>rsync --bwlimit=1000 fromfile tofile</code>	Locally copy with rate limit. It's like nice for I/O
	<code>rsync -az -e ssh --delete ~/public_html/ remote.com:'~/public_html'</code>	Mirror web site (using compression and encryption)
	<code>rsync -auz -e ssh remote:/dir/ . && rsync -auz -e ssh . remote:/dir/</code>	Synchronize current directory with remote one
ssh (Secure SHell)		
	<code>ssh \$USER@\$HOST command</code>	Run command on \$HOST as \$USER (default command=shell)
•	<code>ssh -f -Y \$USER@\$HOSTNAME xeyes</code>	Run GUI command on \$HOSTNAME as \$USER
	<code>scp -p -r \$USER@\$HOST: file dir/</code>	Copy with permissions to \$USER's home directory on \$HOST
	<code>scp -c arcfour \$USER@\$LANHOST: bigfile</code>	Use faster crypto for local LAN. This might saturate GigE
	<code>ssh -g -L 8080:localhost:80 root@\$HOST</code>	Forward connections to \$HOSTNAME:8080 out to \$HOST:80
	<code>ssh -R 1434:imap:143 root@\$HOST</code>	Forward connections from \$HOST:1434 in to imap:143
	<code>ssh-copy-id \$USER@\$HOST</code>	Install public key for \$USER@\$HOST for password-less log in
networking (Note ifconfig, route, mii-tool, nslookup commands are obsolete)		
	<code>ethtool eth0</code>	Show status of ethernet interface eth0
	<code>ethtool --change eth0 autoneg off speed 100 duplex full</code>	Manually set ethernet interface speed
	<code>iw dev wlan0 link</code>	Show link status of wireless interface wlan0
	<code>iw dev wlan0 set bitrates legacy-2.4 1</code>	Manually set wireless interface speed
•	<code>iw dev wlan0 scan</code>	List wireless networks in range
•	<code>ip link show</code>	List network interfaces
	<code>ip link set dev eth0 name wan</code>	Rename interface eth0 to wan
	<code>ip link set dev eth0 up</code>	Bring interface eth0 up (or down)
•	<code>ip addr show</code>	List addresses for interfaces

	ip addr add 1.2.3.4/24 brd + dev eth0	Add (or del) ip and mask (255.255.255.0)
•	ip route show	List routing table
	ip route add default via 1.2.3.254	Set default gateway to 1.2.3.254
•	ss -tupl	List internet services on a system
•	ss -tup	List active connections to/from system
	host pixelbeat.org	Lookup DNS ip address for name or vice versa
•	hostname -i	Lookup local ip address (equivalent to host `hostname`)
•	whois pixelbeat.org	Lookup whois info for hostname or ip address

windows networking (Note samba is the package that provides all this windows specific networking support)

	smbtree	Find windows machines. See also findsmb
	nmblookup -A 1.2.3.4	Find the windows (netbios) name associated with ip address
	smbclient -L windows_box	List shares on windows machine or samba server
•	mount -t smbfs -o fmask=666,guest //windows_box/share /mnt/share	Mount a windows share
	echo 'message' smbclient -M windows_box	Send popup to windows machine (off by default in XP sp2)

text manipulation (Note sed uses stdin and stdout. Newer versions support inplace editing with the -i option)

	sed 's/string1/string2/g'	Replace string1 with string2
	sed 's/(.*\1)\1/\2/g'	Modify anystring1 to anystring2
	sed '/^ *#/d; /^ *\$/d'	Remove comments and blank lines
	sed ':a; /\\$\N; s/\n//; ta'	Concatenate lines with trailing \
	sed 's/[\t]*\$//'	Remove trailing spaces from lines
	sed 's/\\([`"\$`\]\\)/\\\\\\1/g'	Escape shell metacharacters active within double quotes
•	seq 10 sed "s/^/ /; s/ *\(\.\{7,\}\)/\1/"	Right align numbers
•	seq 10 sed p paste - -	Duplicate a column
	sed -n '1000{p;q}'	Print 1000th line
	sed -n '10,20p;20q'	Print lines 10 to 20
	sed -n 's/.*<title>\(.*)</title>.*/\1/ip;T;q'	Extract title from HTML web page
	sed -i 42d ~/.ssh/known_hosts	Delete a particular line
	sort -t. -k1,1n -k2,2n -k3,3n -k4,4n	Sort IPV4 ip addresses
•	echo 'Test' tr '[:lower:]' '[:upper:]'	Case conversion
•	tr -dc '[:print:]' < /dev/urandom	Filter non printable characters
•	tr -s '[:blank:]' '\t' </proc/diskstats cut -f4	cut fields separated by blanks
•	history wc -l	Count lines
•	seq 10 paste -s -d ''	Concatenate and separate line items to a

set operations (Note you can export LANG=C for speed. Also these assume no duplicate lines within a file)	
sort file1 file2 uniq	Union of unsorted files
sort file1 file2 uniq -d	Intersection of unsorted files
sort file1 file1 file2 uniq -u	Difference of unsorted files
sort file1 file2 uniq -u	Symmetric Difference of unsorted files
join -t'\0' -a1 -a2 file1 file2	Union of sorted files
join -t'\0' file1 file2	Intersection of sorted files
join -t'\0' -v2 file1 file2	Difference of sorted files
join -t'\0' -v1 -v2 file1 file2	Symmetric Difference of sorted files
math	
• echo '(1 + sqrt(5))/2' bc -l	Quick math (Calculate φ). See also bc
• seq -f '4/%g' 1 2 99999 paste -sd-+ bc -l	Calculate π the unix way
• echo 'pad=20; min=64; (100*10^6)/((pad+min)*8)' bc	More complex (int) e.g. This shows max FastE packet rate
• echo 'pad=20; min=64; print (100E6)/((pad+min)*8)' python	Python handles scientific notation
• echo 'pad=20; plot [64:1518] (100*10**6)/((pad+x)*8)' gnuplot -persist	Plot FastE packet rate vs packet size
• echo 'obase=16; ibase=10; 64206' bc	Base conversion (decimal to hexadecimal)
• echo \$((0x2dec))	Base conversion (hex to dec) ((shell arithmetic expansion))
• units -t '100m/9.58s' 'miles/hour'	Unit conversion (metric to imperial)
• units -t '500GB' 'GiB'	Unit conversion (SI to IEC prefixes)
• units -t '1 googol'	Definition lookup
• seq 100 paste -s -d+ bc	Add a column of numbers. See also add and funcpy
calendar	
• cal -3	Display a calendar
• cal 9 1752	Display a calendar for a particular month year
• date -d fri	What date is it this friday. See also day
• [\$(date -d '12:00 today +1 day' +%d) = '01'] exit	exit a script unless it's the last day of the month
• date --date='25 Dec' +%A	What day does xmas fall on, this year
• date --date='@2147483647'	Convert seconds since the epoch (1970-01-01 UTC) to date
• TZ='America/Los_Angeles' date	What time is it on west coast of US (use tzselect to find TZ)
• date --date='TZ="America/Los_Angeles" 09:00 next Fri'	What's the local time for 9AM next Friday on west coast US
locales	
• printf "%'d\n" 1234	Print number with thousands grouping appropriate to locale

• <code>BLOCK_SIZE='\1 ls -l</code>	Use locale thousands grouping in ls. See also l
• <code>echo "I live in `locale territory`"</code>	Extract info from locale database
• <code>LANG=en_IE.utf8 locale int_prefix</code>	Lookup locale info for specific country. See also ccodes
• <code>locale -kc \$(locale sed -n 's/\(\LC_\.\{4,\}\)=.*/\1/p') less</code>	List fields available in locale database

recode (Obsoletes iconv, dos2unix, unix2dos)

• <code>recode -l less</code>	Show available conversions (aliases on each line)
<code>recode windows-1252.. file_to_change.txt</code>	Windows "ansi" to local charset (auto does CRLF conversion)
<code>recode utf-8/CRLF.. file_to_change.txt</code>	Windows utf8 to local charset
<code>recode iso-8859-15..utf8 file_to_change.txt</code>	Latin9 (western europe) to utf8
<code>recode ../b64 < file.txt > file.b64</code>	Base64 encode
<code>recode /qp.. < file.qp > file.txt</code>	Quoted printable decode
<code>recode ..HTML < file.txt > file.html</code>	Text to HTML
• <code>recode -lf windows-1252 grep euro</code>	Lookup table of characters
• <code>echo -n 0x80 recode latin-9/x1..dump</code>	Show what a code represents in latin-9 charmap
• <code>echo -n 0x20AC recode ucs-2/x2..latin-9/x</code>	Show latin-9 encoding
• <code>echo -n 0x20AC recode ucs-2/x2..utf-8/x</code>	Show utf-8 encoding

CDs

<code>gzip < /dev/cdrom > cdrom.iso.gz</code>	Save copy of data cdrom
<code>mkisofs -V LABEL -r dir gzip > cdrom.iso.gz</code>	Create cdrom image from contents of dir
<code>mount -o loop cdrom.iso /mnt/dir</code>	Mount the cdrom image at /mnt/dir (read only)
<code>wodim dev=/dev/cdrom blank=fast</code>	Clear a CDRW
<code>gzip -dc cdrom.iso.gz wodim -tao dev=/dev/cdrom -v -data -</code>	Burn cdrom image (use --prcap to confirm dev)
<code>cdparanoia -B</code>	Rip audio tracks from CD to wav files in current dir
<code>wodim -v dev=/dev/cdrom -audio -pad *.wav</code>	Make audio CD from all wavs in current dir (see also cdrdao)
<code>oggenc --tracknum=\$track track.cdda.wav -o track.ogg</code>	Make ogg file from wav file

disk space

• <code>ls -lSr</code>	Show files by size, biggest last
• <code>du -s * sort -k1,1rn head</code>	Show top disk users in current dir. See also dutop
• <code>du -hs /home/* sort -k1,1h</code>	Sort paths by easy to interpret disk usage
• <code>df -h</code>	Show free space on mounted filesystems
• <code>df -i</code>	Show free inodes on mounted filesystems
• <code>fdisk -l</code>	Show disks partitions sizes and types (run as root)

• rpm -q -a --qf '%10{SIZE}\t%{NAME}\n' sort -k1,1n	List all packages by installed size (Bytes) on rpm distros
• dpkg-query -W -f='\${Installed-Size;10}\t\${Package}\n' sort -k1,1n	List all packages by installed size (KBytes) on deb distros
• dd bs=1 seek=2TB if=/dev/null of=ext3.test	Create a large test file (taking no space). See also truncate
• > file	truncate data of file or create an empty file
monitoring/debugging	
• tail -f /var/log/messages	Monitor messages in a log file
• strace -c ls >/dev/null	Summarise/profile system calls made by command
• strace -f -e open ls >/dev/null	List system calls made by command
• strace -f -e trace=write -e write=1,2 ls >/dev/null	Monitor what's written to stdout and stderr
• ltrace -f -e getenv ls >/dev/null	List library calls made by command
• lsof -p \$\$	List paths that process id has open
• lsof ~	List processes that have specified path open
• tcpdump not port 22	Show network traffic except ssh. See also tcpdump_not_me
• ps -e -o pid,args --forest	List processes in a hierarchy
• ps -e -o pcpu,cpu,nice,state,cputime,args --sort pcpu sed '/^ 0.0 /d'	List processes by % cpu usage
• ps -e -orss=,args= sort -b -k1,1n pr -TW\$COLUMNS	List processes by mem (KB) usage. See also ps_mem.py
• ps -C firefox-bin -L -o pid,tid,pcpu,state	List all threads for a particular process
• ps -p 1,\$\$ -o etime=	List elapsed wall time for particular process IDs
• watch -n.1 pstree -Uacp \$\$	Display a changing process subtree
• last reboot	Show system reboot history
• free -m	Show amount of (remaining) RAM (-m displays in MB)
• watch -n.1 'cat /proc/interrupts'	Watch changeable data continuously
• udevadm monitor	Monitor udev events to help configure rules
system information	
• uname -a	Show kernel version and system architecture
• head -n1 /etc/issue	Show name and version of distribution
• cat /proc/partitions	Show all partitions registered on the system
• grep MemTotal /proc/meminfo	Show RAM total seen by the system
• grep "model name" /proc/cpuinfo	Show CPU(s) info
• lspci -tv	Show PCI info

• lsusb -tv	Show USB info
• mount column -t	List mounted filesystems on the system (and align output)
• grep -F capacity: /proc/acpi/battery/BAT0/info	Show state of cells in laptop battery
# dmidecode -q less	Display SMBIOS/DMI information
# smartctl -A /dev/sda grep Power_On_Hours	How long has this disk (system) been powered on in total
# hdparm -i /dev/sda	Show info about disk sda
# hdparm -tT /dev/sda	Do a read speed test on disk sda
# badblocks -s /dev/sda	Test for unreadable blocks on disk sda
interactive	
• readline	Line editor used by bash, python, bc, gnuplot, ...
• screen	Virtual terminals with detach capability, ...
• mc	Powerful file manager that can browse rpm, tar, ftp, ssh, ...
• gnuplot	Interactive/scriptable graphing
• links	Web browser
• xdg-open .	open a file or url with the registered desktop application

COPA - Using Word Processing Software

MS WORD 2010 THEORY

Objectives : At the end of this lesson you shall be able to

- state what is MS Office
- brief what is MS Word and starting steps
- explain various screen blocks of MS Word
- explain procedures to create, save, print a document!

Microsoft office is a application software package introduced by Microsoft Corporation. MS Office consists of the following popular packages :

- Microsoft Word
- Microsoft Excel
- Microsoft Powerpoint
- Microsoft Access
- Microsoft Outlook

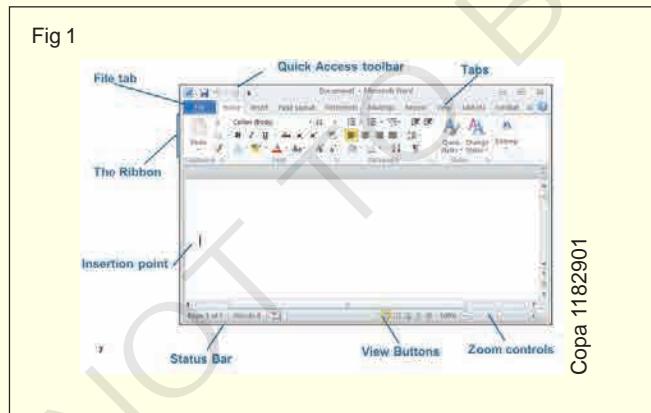
Word 2010

It is a word processor package that helps to create and edit a document. It is the most known word processor of nowadays. It makes professional looking documents by providing a comprehensive set of tools for creating and formatting a document, memos, letters, reports, brochures, business documents and even internet web pages.

Starting Word

Click on the Start > (windows Logo) All programmes > Microsoft office > Microsoft Word. MS Word opens along with a default blank document with default settings page layout.

Default page layout (Fig 1)

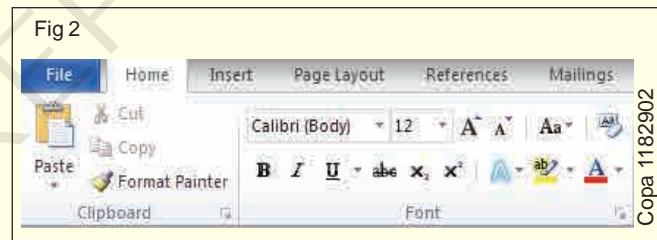


Unlike previous version, MS Office 2010 has a common set of features for all the components. It makes it easy to understand and common utilities to remain available on all the packages of MS Office including Word, Excel, PowerPoint, Access, etc. Many features are redesigned so as enabling the diverted users of other packaged can afford with. These enhancements and utilities in Word 2010 are described as below.

The Ribbon tab of Word has eight major parts viz. File, Home, Insert, Page Layout, References, Mailings, Review and View.

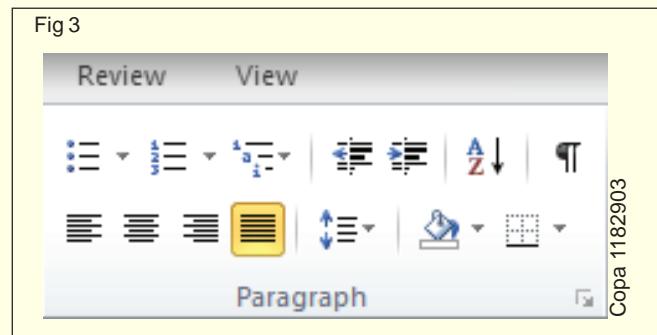
File tab of the ribbon is useful to create a new file, a blank or a template page as required. Ctrl + N always create a new blank document in Office Package. The documents so created can be saved as default word document or given compatible format. Furthermore, permission for accessing a documents can be fixed, share a document on a network, even manage a document to be compatible with previous and external versions. Print option makes the document to get a hard copy or a pdf format according to the installed printer features. Recently opened files can also be viewed to find it easy to work again. Help on word is there in the same tab.

Home Tab (Fig 2)



Home tab has clipboard options including cut, copy, paste and paste special. The Font block has all the options of setting fonts, size, superscript, subscript, bold, italic, underline, font colors, etc.

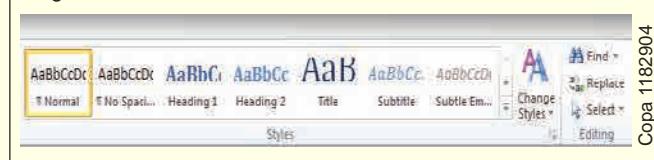
Paragraph Group (Fig 3)



Paragraph group creates the formatting paragraphs with alignments left, right, center, justify and indentations, para and line spacing allows to create pagination options and exceptions of formatting.

Styles and Editing Group (Fig 4)

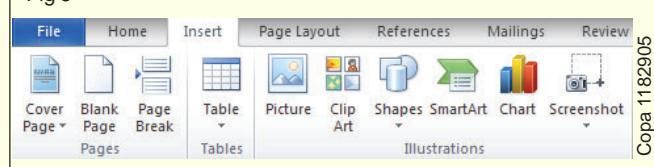
Fig 4



The Styles group allows preformatted text styles like Heading styles, Paragraph styles, Subtitles, etc. Custom styles can also be stored with altered specifications. The Editing group used to select specific area, find and / or replace option in a specified area in text.

The Insert tab has 7 groups. Pages, Tables and Illustrations blocks (Fig 5)

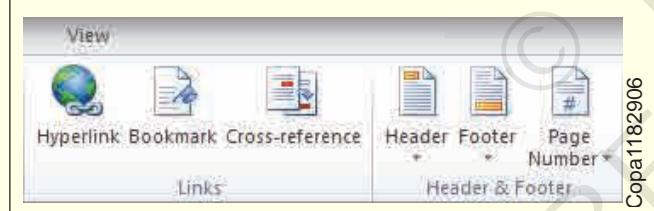
Fig 5



Page group makes cover pages, blank pages insertion and page breaks. Tables group helps to insert a table in a text document with ready specified row-column set or a customized table format. Illustration group inserts pictures from external sources, cliparts, shapes, smart art, charts of data, even screenshot into the text document.

Links and Header/Footer Groups (Fig 6)

Fig 6

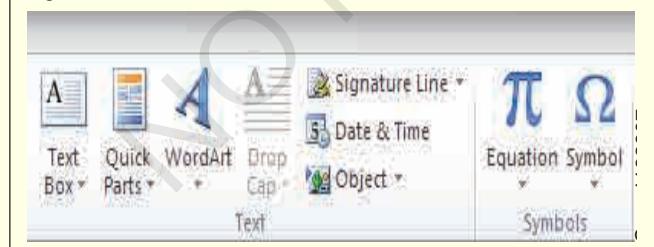


Links group creates hyperlink on texts, bookmarks and cross references in a document.

The header and footer group inserts header, footer to be appearing on every page and page numbers to display as to placement area.

Text and Symbols block (Fig 7)

Fig 7

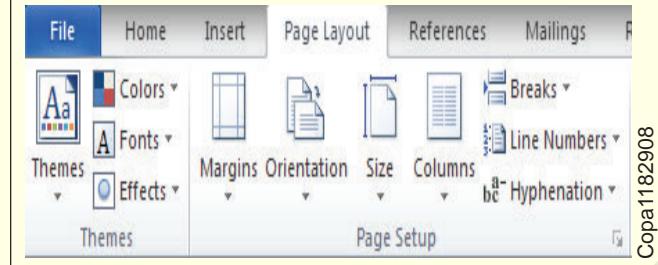


Text group allows creating text box, quick parts likely to create brochures, designed text as WordArt, paragraph styles, a signature line, date and time and an object insert option.

Symbols group inserts symbols of equations like math equations or symbols like currency, math symbols, etc.

Themes and Page Setup Group (Fig 8)

Fig 8

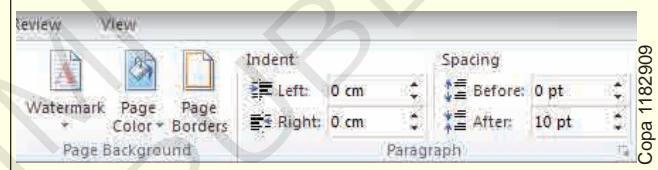


Page Layout tab has five major groups. Themes block creates predefined template setup using themes on documents. Even new themes customized can be created and saved for future use.

Page setup group has features on margin around, page orientation i.e. vertical or horizontal, paper size, columns to display, breaks, line numbers and hyphenation.

Page Background and Paragraph Group (Fig 9)

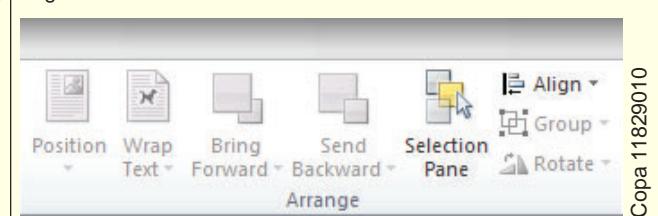
Fig 9



Page Background group creates a watermark, page background color and page borders. Paragraph block has specified options already discussed in Home -> Paragraph group.

Arrange Group (Fig 10)

Fig 10



The arrange group creates position of objects, text elements, text wrapping, arranging objects, aligning of objects, grouping and transformations.

References tab has six blocks.

TOC and Footnotes Group (Fig 11)

Fig 11

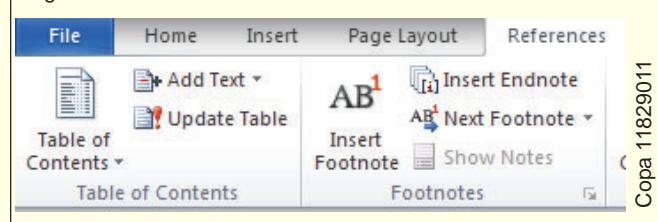
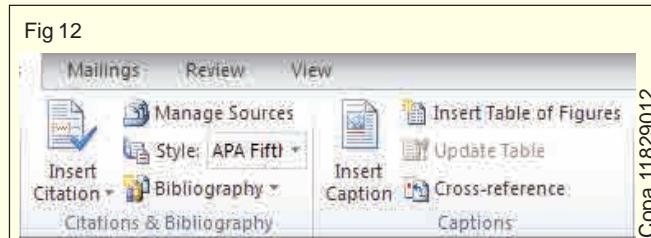


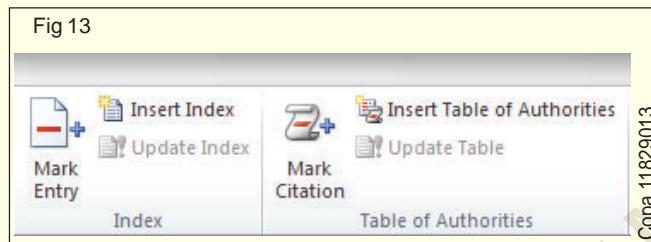
Table of contents creates the TOC of a particular publication document automatically and upon modifications updatable. Footnotes tab creates footnotes of a page, block area which includes explanatory phrases or references. Endnote makes the note at the end of the document.

Citations, Bibliography and Captions Group (Fig 12)



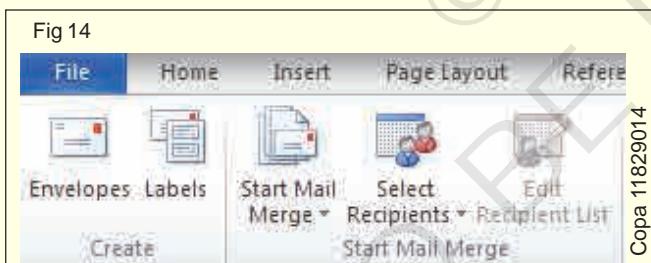
Citation and Bibliography makes an external source as citation, reference tables and credits of authors related to the publication. Captions tab inserts a caption, table of figures, cross references inside a text.

Index and Table of Authorities Group (Fig 13)



Index enters a value on a main topic or sub topic and create the index accordingly. Table of authorities has the citation mark includes the list of the cases, statutes and other authorities cited in the document.

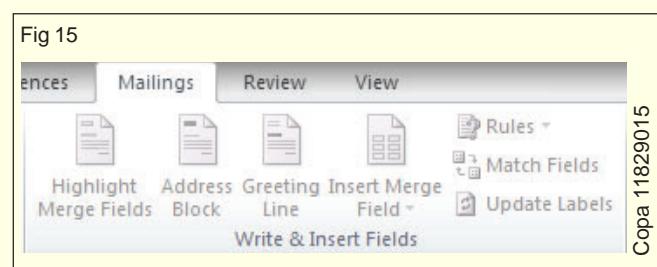
Create and Mailmerge Groups (Fig 14)



Mailings tab used to create mailing of a letter or order using common mailing features. Create block makes envelopes print using predefined formats as well as custom formats. Labels can also be printed to which the delivery address is mentioned.

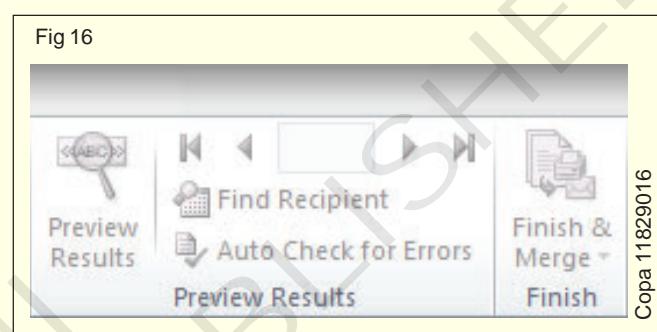
Mail merge block is used to create mail merge of a letter, email. If a common letter is created and to be sent to many recipients, this option allows to create an Address block where the recipient's info should appear. Even selective recipients can be listed as well as a new recipient list can be created.

Write and Insert fields Group (Fig 15)



While writing a letter, it has many parts, which this mail merge creates fields of Address Block, Greetings Line, Merged field of content, labels, etc.

Preview Results and Finish Groups (Fig 16)



On successful completion of mail merge, it can be previewed and checked for errors for each recipients and edit accordingly. Upon validation the mail merge gets finished.

Proofing and Language Groups (Fig 17)



Review tab here is used for the various document areas to be reviewed. Proofing makes easier to check Spelling and Grammar of a document. Accidental errors can be corrected here. Research refines the search not only inside the document, but also the referenced documents. Thesaurus identifies the completeness of the document using the standard language compatibility of the region like English UK or English USA etc. Word count shows the analysis of the document as total words, total characters, with and without spaces, no. of paragraphs, lines, etc. Here Text blocks can also be included for such analysis.

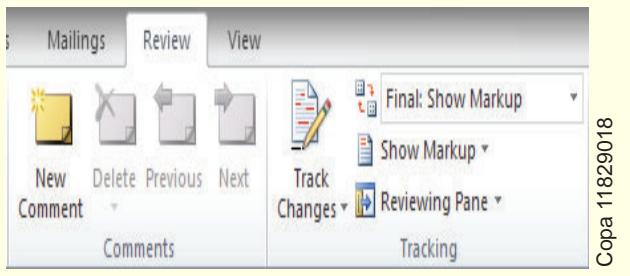
Language Block helps to translate a page into the installed other languages and to change the proofing language. It requires the direct translator service from Microsoft Online.

Comments & Tracking Group (Fig 18)

Comments group inserts a comment for a specific paragraph or text block.

Tracking group finds the changes made on a document by other authors in a protected mode. For example, if a document is created by user X and edited by user Y, is tracked separately with Track marks.

Fig 18



Changes, Compare and Protect Groups (Fig 19)

Changes hence made on the documents can either then accepted or rejected. More than a document can be compared for similarity using Compare. Also the document can be protected from editing by other authors.

Fig 19



Creating a file, save and other options

Objectives: At the end of this lesson you shall be able to

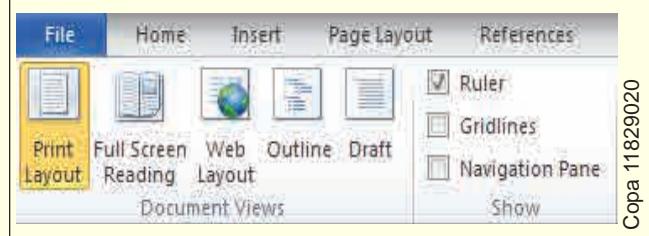
- explain how to create a new document, save and print
- state how to edit, format text and document styles
- brief using tables inside word document with data
- explain how to create styles in a document and save for future use
- explain Few unique features of Word 2010
- brief the mail merge processing

File Info view (Fig 1): Word 2010 new document can be created as usual with Ctrl + N or through File menu New option. Basically the file created is saved as a word document extended format as docx in word file.

Save options in Word (Fig 2): It can be saved using save as option in any compatible format or old versions of office, like 2003 or earlier versions. Main utility of the word software is the creation of word processing documents. It may be any of a format like publication, letter, brochure, etc. Word supports all type of formatting to design a text based presentation. Also it supports

Document views and show Groups (Fig 20)

Fig 20



View tab shows the way of displaying the word document. Document views have a Print Layout, a common view of Word, Full Screen Reading, minimises the tabs and ribbons to disappear and easy to read, Web Layout, which previews a html compatible view, Outline, views the basic version of document in mere text mode and Draft mode for a text editing mode.

The Rulers, Gridlines and Navigation Pane can be shown or hidden according to user preferences.

Zoom / Window / Macro Groups (Fig 21)

Fig 21



Also the view of the page can be Zoomed to full page, two pages view, 100% of the document and custom view. To make easy editing a document the window can be split into two, a new window for a document to cut paste, etc. can be created and all open word documents can be arranged for view. While formatting the documents, for repetition of commands the Macro option can be used. It uses the Record option to store the set of commands and repeat it again on other part of document or on another document on a single click.

Fig 1

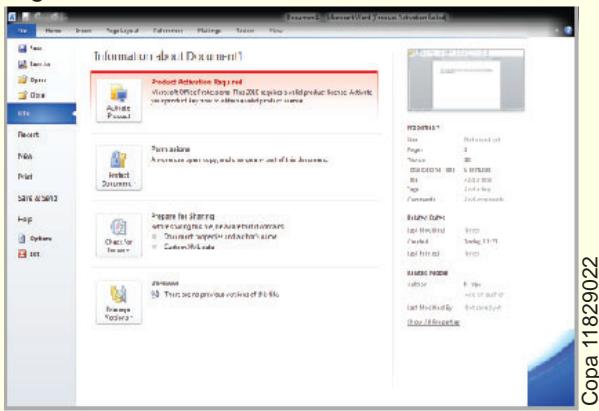


Fig 2

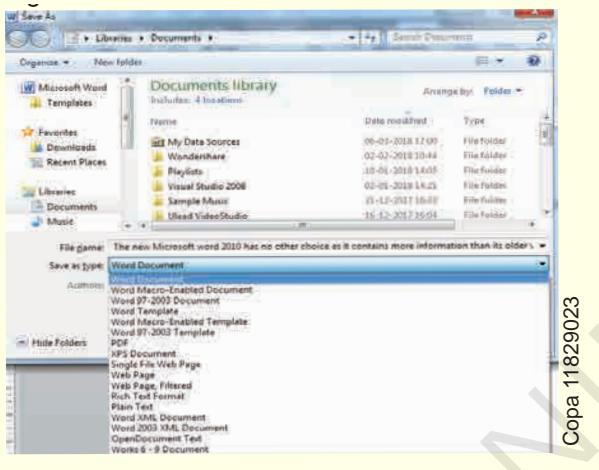
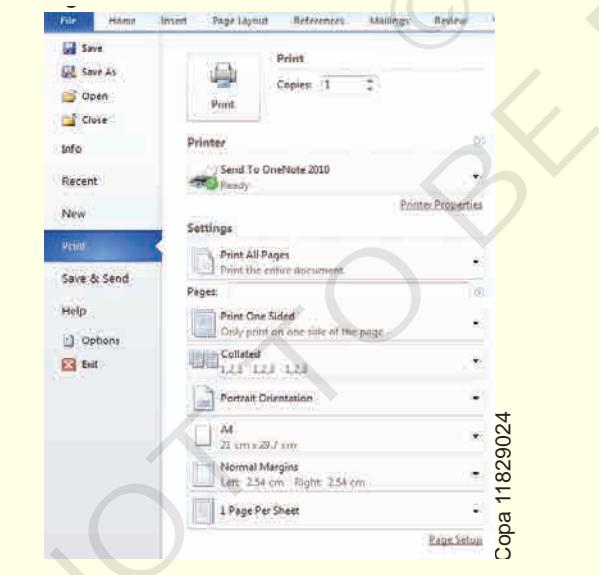


Fig 3



Basic concepts to be noted while using Word are Text properties. It includes Font type, Font size, Text Color, and usual decorations of text. Also creating of paragraphs styles are to be kept in mind. Paragraph alignment has left, right, center and justified settings. Text elements may contain items like ordered list, unordered list, subsection lists. They are found there paragraph formatting block of Home Tab. Indenting of text for creating Quotes is also there inside the same tab. According to the page

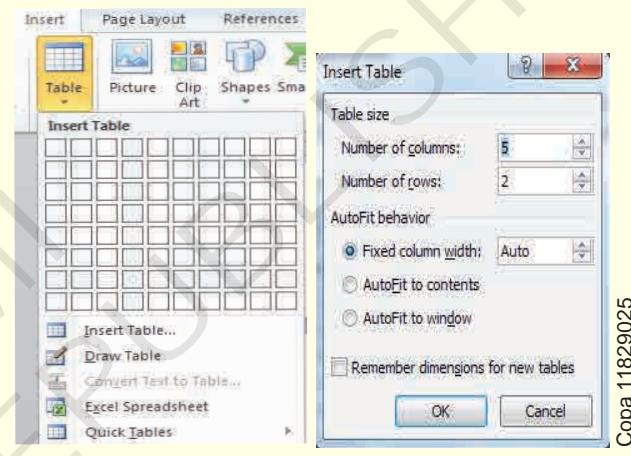
size, line spacing and paragraph spacing can be adjusted, like before and after paragraph spaces, line heights, etc. The Styles can be predefined to use as ready to put on places where it required. Standard templates are available but it allows to create custom styles too.

Tables can be inserted for various utilities. Normal table has adjustable width and columns so it can simply inserted with Insert > Table directly.

Table insert wizard options (Fig 4)

In case of custom sized table is required, it can be created using insert table / draw table options. Insert table allows custom values for columns, width and other properties. Draw table creates custom table using a pen drawing tool through which new table can be drawn according to the available text contents.

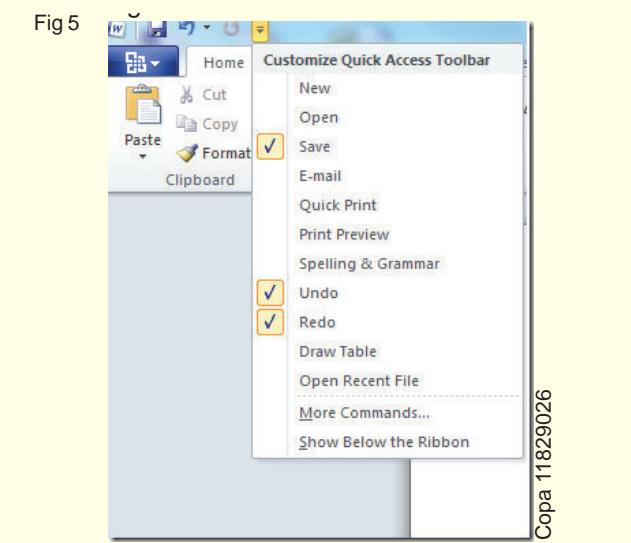
Fig 4



Customizable Quick Access Tool Bar

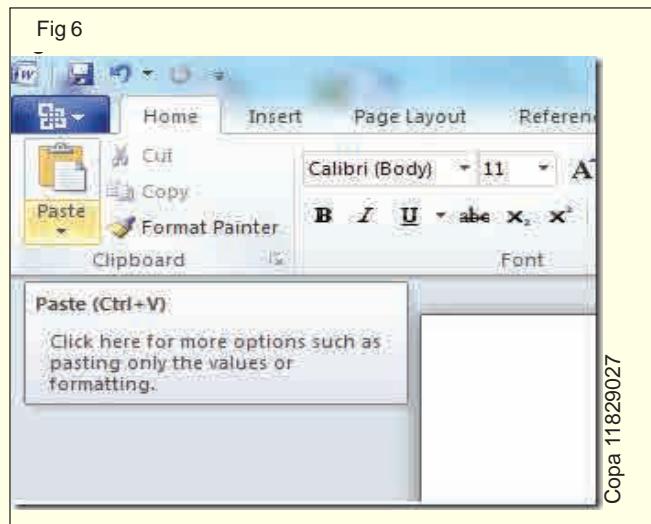
Word 2010's Quick Access Toolbar displays all the commonly used options. It is located in the top left side corner of the application window, near the office button. By default it displays the following three options, Save, Undo and Redo, but is customizable and you may easily add more options to it. (Fig 5)

Fig 5



Paste Preview

It happens with most users that after copying and pasting something into their document, they need to undo the some changes. Word 2010 has made it easy for users, now you may eliminate this unnecessary step by using the paste preview option. It allows users to paste only the values or the formatting. (Fig 6)

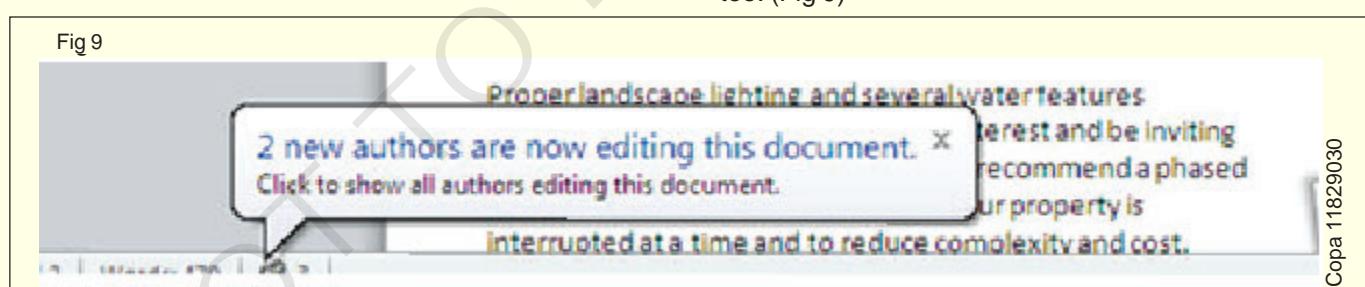


Navigation Pane

In the previous versions of Microsoft Office, one has to use the Ctrl+F hotkey to find any word or phrase from within a document. Word 2010 has added a new magic to this option, Ctrl+F now summons a Navigation Pane that appears on the left side of the document. You will see the three views available by clicking on their respective tabs, the Heading View, Thumbnail Page View, and the Search Result View. (Fig 7)

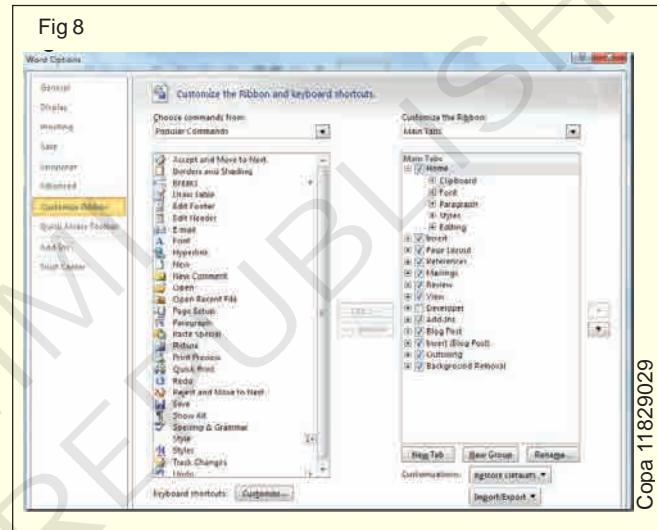
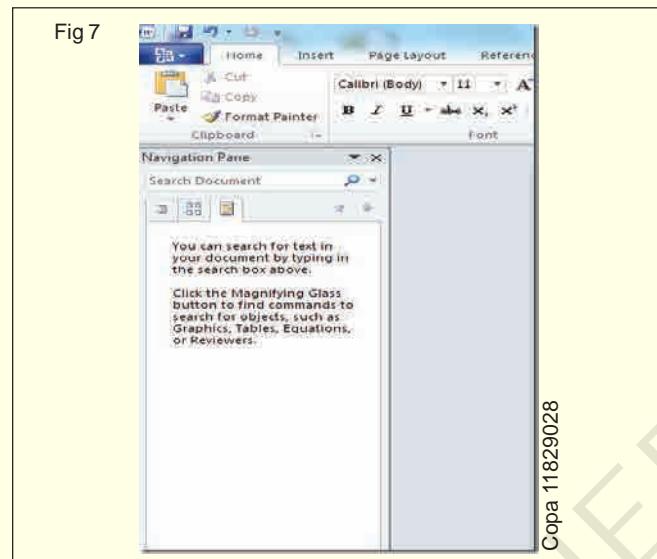
Customizable Ribbon Button

Apparently the Ribbon button in Word 2010 looks like the one in Word 2007. But there is one big addition, you may customize the word 2010's Ribbon button. In order to customize the Ribbon button navigate to the following option Office Button > Word Option > Customize Ribbon. (Fig 8)



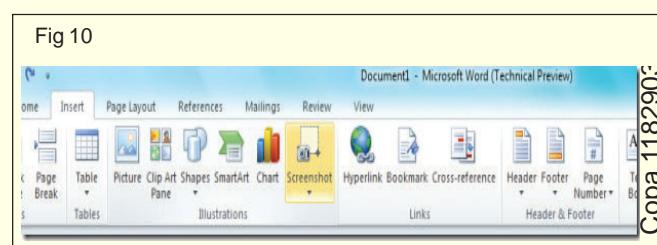
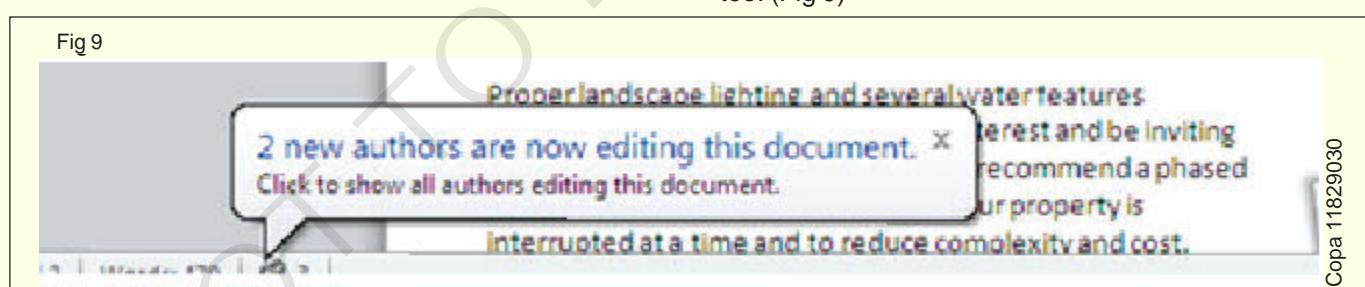
Screen Capture Tool

Word 2010 includes a feature called Screen Capturing, now there is no need to use a third party or additional tool to capture a screenshot in order to use it in Word, just simply use Word 2010's built in tool to capture any area of the screen. A Screenshot may be taken by navigating to the following option Insert > Screenshot. (Fig 10)



Improved Collaborations

Microsoft Word 2010 has a new feature called co-authoring. It allows more than one authors to edit a document at the same time. Word 2010 tells you how many authors are editing the document and their changes can be viewed too. (Fig 9)



Artistic Effects

In Word 2010 users can now apply a number of snazzy artistic effects to the pictures. In order to add the artistic effects to your document, Navigate to the following option Insert > Illustrations > Picture. Then browse and select the picture you want. Once the picture is added to your document, then the Picture Tools contextual tab is displayed and you will be able to see the new Artistic Effects drop down button over here. (Fig 11)

Fig 11



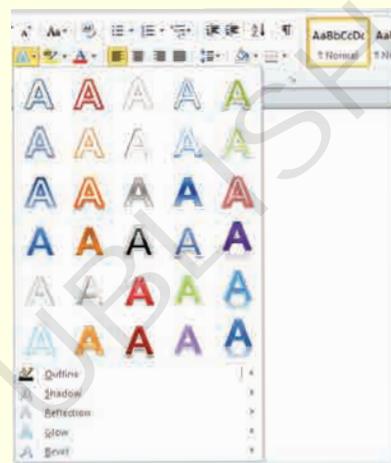
Copa 11829032

Fig 13



Copa 11829034

Fig 14

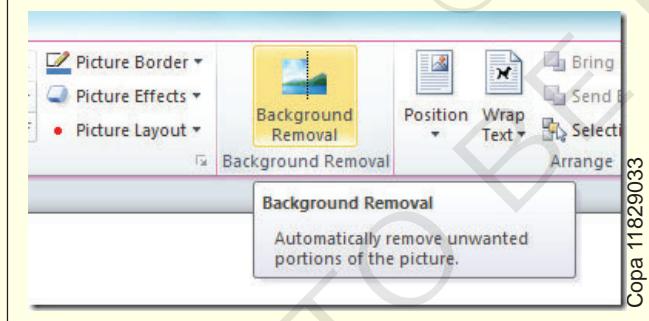


Copa 11829035

The Background Removal Option

Office 2010 has an awesome option by the name of Background Removal, which simply removes the background of any image. Yes, you don't need Photoshop anymore to remove the background. First insert the picture in your Word document from the Insert > Picture option. Then locate the Background Removal tool and get rid of the background. (Fig 12)

Fig 12

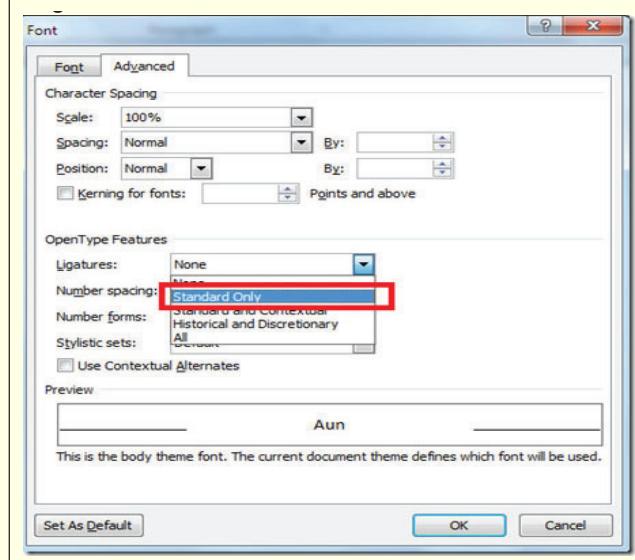


Copa 11829033

Ligatures

You might have heard about Ligatures. They make the fonts look fancy and they are also used to keep letters separate allowing you to search the text as if the font were regular. Its true that not all fonts support ligatures, but a large variety of the fonts supports them. You may enable them from Font Preferences > advanced, then select the standard only option in the ligatures drop down box. (Fig 15)

Fig 15



Copa 11829036

Office Backstage

The Office Backstage is a new concept, it is the enhanced form of the plain old office button and provides a much user-friendly menu. It helps users to manage documents, presentations, or spreadsheets at a greater level. (Fig 13)

Interesting feature, right? Read more about it here.

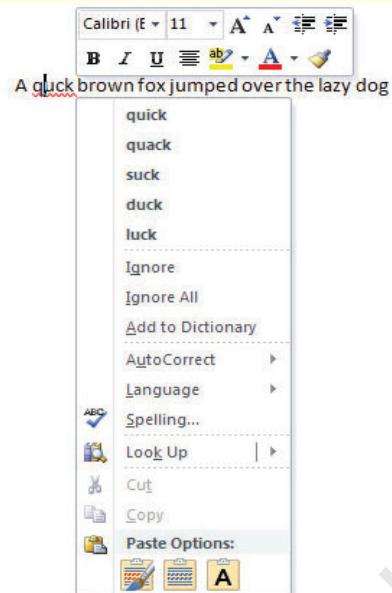
New Art Effects in WordArt

Just like other features, WordArt has been updated with new colorful art effects. Select the text, then click Word Art and a list of all the available options will be displayed. (Fig 14)

Improved Spell Checks

Word 2010 has added some new features to its spell checker, now it will detect the mistake and suggest changing the sentence.

Fig 16



Shortcut keys in Word 2010

Objectives: At the end of the lesson you shall be able to

- Learn the shortcut keys in MS Word.

CTRL+SHIFT+A	converts the selected text to capital letters or vice versa
CTRL+SHIFT+F	Displays the Font dialog box.
CTRL+SHIFT+G	Displays the Word Count dialog box.
CTRL+SHIFT+S	Displays the Apply Styles task pane.
ALT+R	Displays the Review tab
ALT+CTRL+1	Apply Heading 1, Similarly ALT + CTRL + 2 will apply heading 2
CTRL+SHIFT+L	Applies Bullets
CTRL+SHIFT+F5	Bookmark
CTRL + B	Bold Text
CTRL + I	Italic Text
CTRL + U	Underline Text
CTRL+PAGE DOWN	Browse Next
CTRL+E	Navigate to the center Paragraph
CTRL+SHIFT+ENTER	Column Break
CTRL+SHIFT+C	Copy Format
ALT+SHIFT+F7	Dictionary
ALT+CTRL+S	Splits the Document
CTRL+SHIFT+D	Double Underline
CTRL+END	End of Document
END	End of line
CTRL+SHIFT+P	Font size select
SHIFT+F5 or ALT+CTRL+Z	Go Back to previous state

CTRL+SHIFT+.	Grow Font
CTRL+]	Grow Font one point
ALT+SHIFT+R	Header Footer Link
CTRL+K	Hyperlink
CTRL+M	Indentation
CTRL+J	Justifies Paragraph
ALT+F8	Inserts Macros
ALT+SHIFT+K	Mail Merge Check
F10	Menu Mode
ALT+F7	Moves to the Next Misspelling
CTRL+H	Replace
CTRL+P	Print
CTRL+SHIFT+F12	Also launches Print
ALT+SHIFT+BACKSPACE	Redo
F12	Save As
CTRL+SHIFT+K	Small Caps
CTRL+SHIFT+S	Style
SHIFT+F7	Thesaurus
ALT+SHIFT+T	Time Field
CTRL+SHIFT+M	Unindent
	-

Insert, format text and paragraphs, Create and configure document sections

Objectives: At the end of this lesson you shall be able to

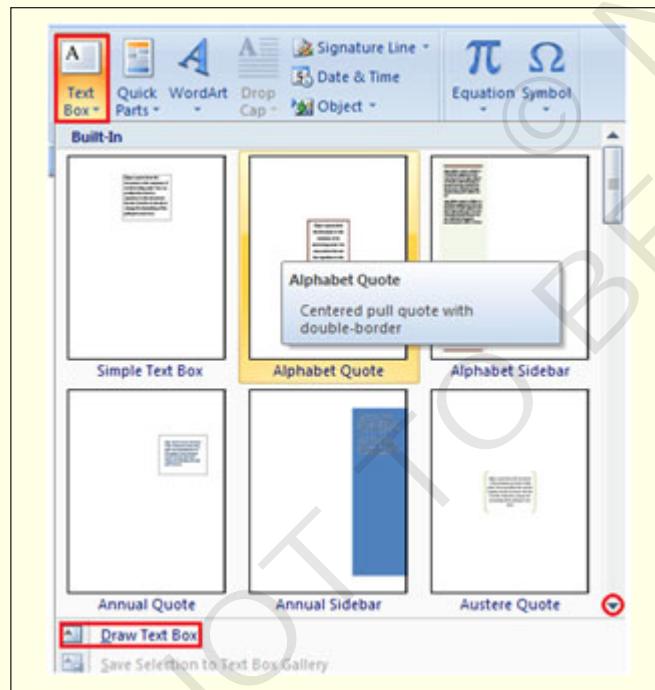
- **insert Text in MS Word**
- **format text**
- **add a section break.**

Insert Text in MS Word

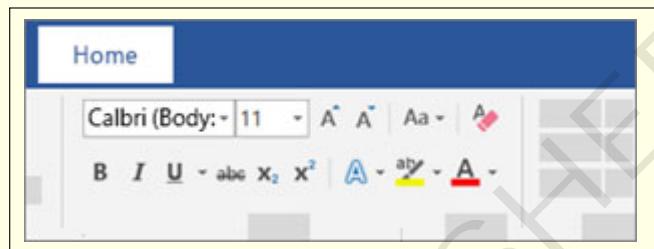
You will see a blinking cursor or insertion point in the text area below the ribbon. Now, as you start typing, the words will appear on the screen in the text area. To change the location of insertion point **press spacebar, Enter or Tab keys**.

The basic steps to insert text or to create a new document in Word are listed below;

- Go to the start menu and look for Microsoft Word icon
- Click the icon to open the Microsoft Word
- You will see a blinking cursor or insertion point in the text area below the ribbon
- Now, as you start typing, the words will appear on the screen in the text area
- To change the location of insertion point press spacebar, Enter or Tab keys



Format text



- 1 Select the text you want to format.

To select a single word, double-click it. To select a line of text, click to the left of it.

- 2 Select an option to change the font, font size, font color, or make the text bold, italic, or underline.

Copy formatting

- 1 Select the text with the formatting you want to copy.
- 2 Click **Format Painter** (represented by a paintbrush icon), and then select the text you want to copy the formatting to.

Tip: Double-click Format Painter if you want to copy the formatting in more than one place.

Add a section break

- 1 Select where you want a new section to begin.
- 2 Go to **Layout > Breaks**.
- 3 Choose the type of section break you want: Next Page Section break starts the new section on the following page. Continuous Section break starts the new section on the same page

Add a section break

- 1 Select where you want a new section to begin.
- 2 Go to Layout > Breaks.

Page Breaks

- Page**
Mark the point at which one page ends and the next page begins.
- Column**
Indicate that the text following the column break will begin in the next column.
- Text Wrapping**
Separate text around objects on web pages, such as caption text from body text.

Section Breaks

- Next Page**
Insert a section break and start the new section on the next page.
- Continuous**
Insert a section break and start the new section on the same page.
- Even Page**
Insert a section break and start the new section on the next even-numbered page.
- Odd Page**
Insert a section break and start the new section on the next odd-numbered page.

3 Choose the type of section break you want:

- **Next Page** - Section break starts the new section on the following page.



- **Continuous** - Section break starts the new section on the same page. This type of section break is often used to change the number of columns without starting a new page.



- **Even Page** - Section break starts a new section on the next even-numbered page.



- **Odd Page** - Section break starts a new section on the next odd-numbered page.



Create, modify tables

Objectives: At the end of this lesson you shall be able to

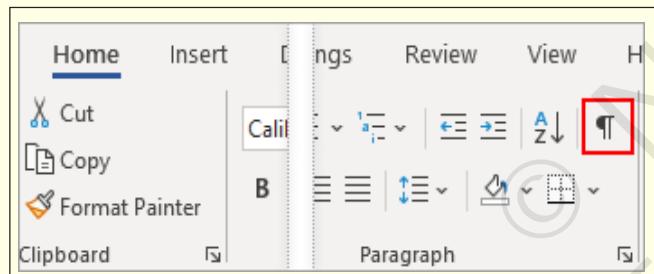
- create tables
- convert text to a table
- split a table.

For a larger table, or to customize a table, select Insert > Table > Insert Table. Tips: If you already have text separated by tabs, you can quickly convert it to a table. Select Insert > Table, and then select Convert Text to Table.

Create tables

Data sets, particularly of numeric data, can often be presented more clearly and efficiently in a table than in a paragraph of text. Tables present large amounts of data, or complex data, in a format that is easier to read and understand by structuring it in rows and columns, which often include headers to explain the purpose or meaning of the data.

To convert text to a table or a table to text, start by clicking the **Show/Hide** paragraph mark on the **Home** tab so you can see how text is separated in your document.



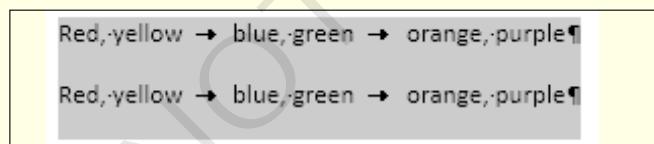
Convert text to a table

- 1 Insert separator characters-such as commas or tabs-to indicate where to divide the text into table columns.

Note: If you have commas in your text, use tabs for your separator characters.

- 2 Use paragraph marks to indicate where you want to begin a new table row.

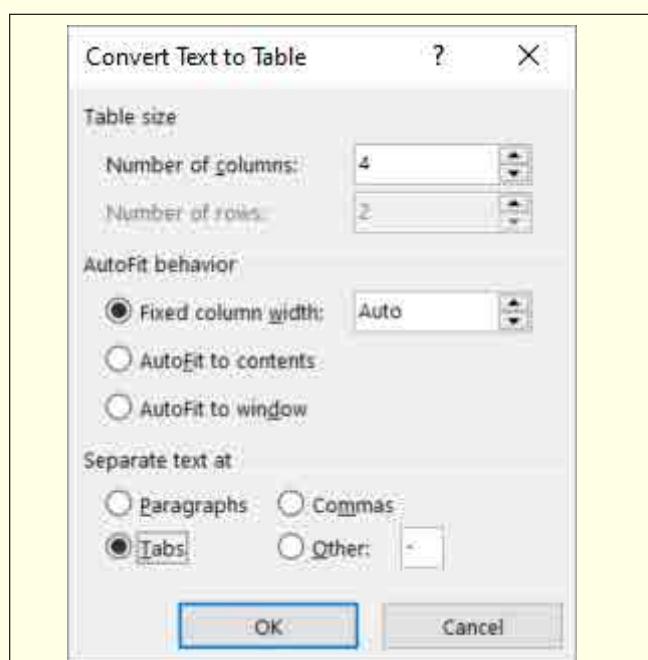
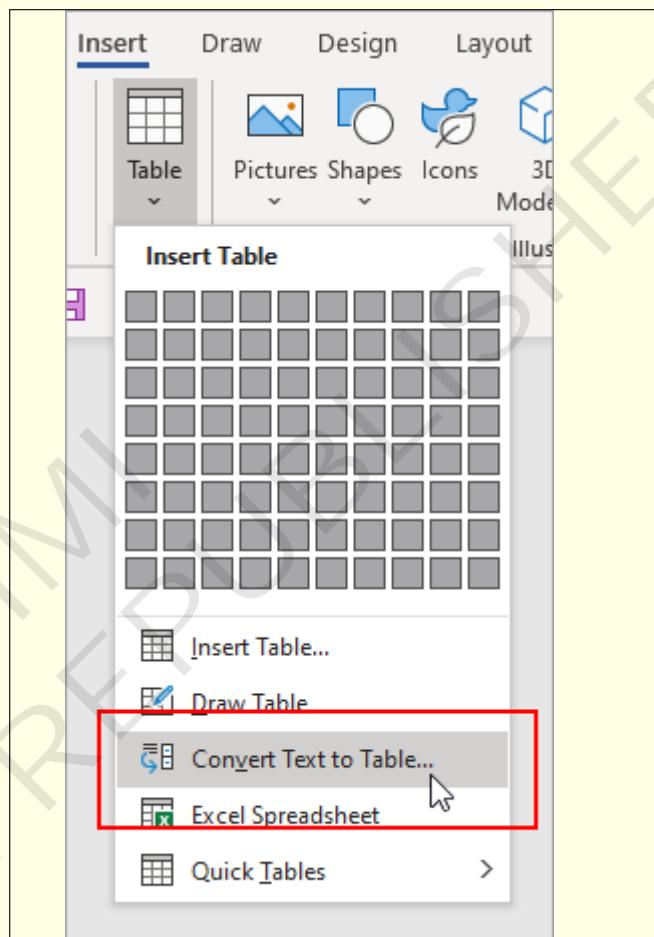
In this example, the tabs and paragraph marks will produce a table with 3 columns and 2 rows:



- 3 Select the text that you want to convert, and then click Insert > Table > Convert Text to Table.

- 4 In the **Convert Text to Table** box, choose the options you want.

Under **Table size**, make sure the numbers match the numbers of columns and rows you want.



- 1 Under **AutoFit behavior**, choose how you want your table to look. Word automatically chooses a width for the table columns. If you want a different column width, choose one of these options:

To do this	Choose this option
Specify a width for all the columns	In the Fixed column width box , type or select a value.
Resize the columns to fit the width of the text in each column	AutoFit to contents
Resize the table automatically in case the width of the available space changes (for example, web layout or landscape orientation)	AutoFit to window

- 2 Under **Separate text at**, choose the separator character you used in the text.
 3 Click **OK**. The text converted to a table should look something like this:

Red, yellow	blue, green	orange, purple
Red, yellow	blue, green	orange, purple

Split a table

- 1 Put your cursor on the row that you want as the first row of your second table. In the example table, it's on the third row.
 2 On the LAYOUT tab, in the Merge group, click Split Table. The table splits into two tables.

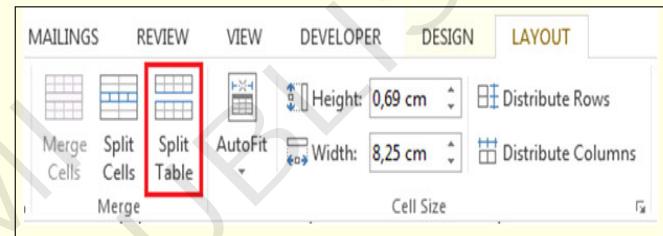
Once you have a table in Word, you might decide to split that table into two or more tables. This way, you can create smaller tables, or add text in between two tables.

- 1 Put your cursor on the row that you want as the first row of your second table. In the example table, it's on the third row.

1	A
2	B
3	C
4	D

When you click inside the table, two new **Table tools** tabs appear on the ribbon: **DESIGN** and **LAYOUT**. These tools are visible only when you are in the table, for example, when you add content to the cells.

- 2 On the **LAYOUT** tab, in the **Merge** group, click **Split Table**.



The table splits into two tables.

You can split the table further, as long as there are multiple rows.

1	A
2	B
3	C
4	D

Create and manage reference elements and tables

Objectives: At the end of this lesson you shall be able to

- manage reference elements
 - customize tables of contents.

Word has its own, basic referencing system available via the 'References' tab on the ribbon. This lets you add references to a document and then create a bibliography at the end of the text. References are stored in a master list, which can be used to add references to further documents.

Note: this system is not as powerful as EndNote, Mendeley or Zotero.

First steps

- 1 Create your document.
 - 2 To insert a citation in the text go to the 'References' tab on the ribbon and click on 'Insert Citation' and 'Add new source'.
 - 3 Select the 'Type of source' and fill in the boxes.
 - 4 Add all the citations to your document.
 - 5 When you have finished, go to the end of your document and click on the 'Bibliography' option. Select from one of the preformatted options or just insert the bibliography to format yourself.
 - 6 To change the style of your references from the default 'APA' style click on the 'Style' list and select another.

Note that the styles available in Word might not match those required for your assignments. Check them carefully and if they don't match you can convert the bibliography to editable text. See the guide below for instruction on doing this.

References in Word are stored on your computer's hard disk. If you want to work on them on another computer, follow these steps to find the file:

- type% APPDATA% to the Start menu search box
 - click the folder **Roaming > Microsoft > Bibliography**
 - the references are saved in the XML file **Sources**

References in Word

Microsoft Word has a simple built-in reference function in the References tab. Here you can manually enter references and refer to them in the text and create a bibliography in the most common styles (APA, Harvard, MLA, etc). This built-in function is good if you have a limited number of references and you don't need to have access to your references on different computers.

On the Ribbon, go to the References. Under the Table of Contents group on the left, click the Table of Contents button. You can choose to insert a default option, or click on Custom Table of Contents... at the bottom of the menu. To insert a custom Table of Contents, select the option from the menu.

After you have created a table of contents in Word, you can customize the way it appears. Customizing your table of contents applies your preferences to your existing table. Like what you see, select **OK**. If you're not happy with the look, just select **Cancel** and all changes are forgotten. Your table of contents will still be there.

To customize your existing table of contents:

- ▶ 1 Go to References > Table of Contents.
 - 2 Select **Custom table of contents**.
 - 3 Use the settings to show, hide, and align page numbers, add or change the tab leader, set formats, and specify how many levels of headings to show. For more info, see Custom table of contents .



COPA - Manage Graphic Elements

Insert, format illustrations and text boxes

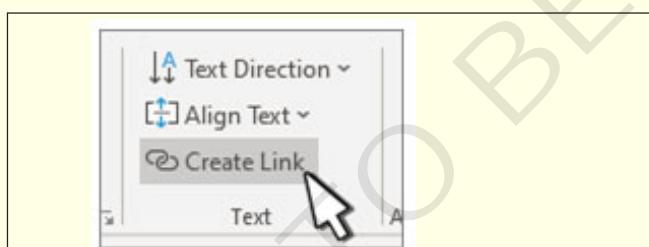
Objectives: At the end of this lesson you shall be able to

- **text box**
- **add text to graphic elements.**

A text box is an object you can add to your document that lets you put and type text anywhere in your file. Text boxes can be useful for drawing attention to specific text and can also be helpful when you need to move text around in your document.

Go to Insert > Text Box, and then select one of the pre-formatted text boxes from the list, select More Text Boxes from Office.com, or select Draw Text Box. If you select Draw Text Box, click in the document, and then drag to draw the text box the size that you want.

- To format the text in the text box, select the text, and then use the formatting options in the **Font** group on the **Home** tab.
- To format the text box itself, use the commands on the **Format** contextual tab, which appears under **Drawing Tools** when you select a text box.
- To position the text box, click it, and then when the pointer becomes a , drag the text box to a new location.
- You can also change or remove a border from a text box or shape.
- If you have multiple text boxes, you can link them together so that text will flow from one box to another. Click one of the text boxes and on the **Format** tab, under **Drawing Tools**, in the **Text** group, click **Create Link**.

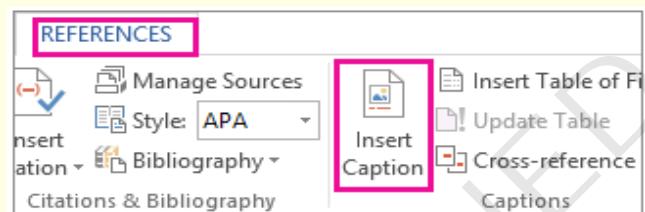


Insert a caption for a picture

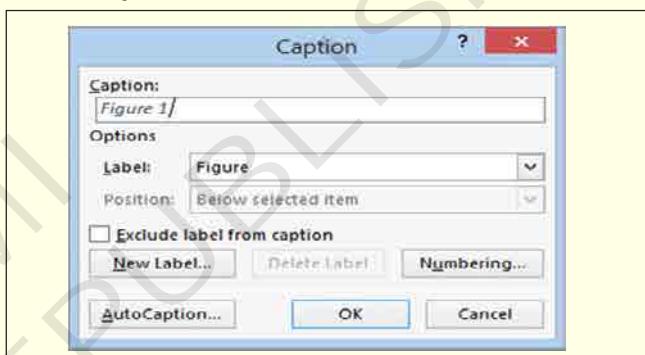
The **Insert Caption** feature in Word makes it easy to systematically add captions to pictures in a document.

In other Office apps, such as PowerPoint, you manually add a text box near the image and then group the text box and image. See the instructions below. If you have multiple pictures in a series, you have to manually number them.

- 1 Click the picture you want to add a caption to.
- 2 Click References > Insert Caption.



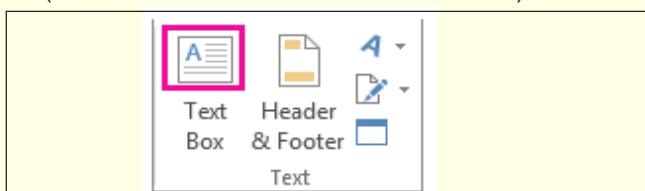
- 3 To use the default label (Figure), type your caption in the **Caption** box.



Tip: You can also create your own default caption label by clicking **New Label**, and then adding your caption in the **Label** box. Word automatically numbers the new labels for you.

All other Office apps

- 1 Use **Insert > Text Box** to draw a box near the picture. (See add a text box for additional details.)



- 2 Click inside the text box and type the text you want to use for a caption.
- 3 Select the text. On the **Home** tab, use the **Font** options to style the caption as you want.
- 4 Use **Ctrl+click** to select the picture and text box, and then on the **Picture Format** tab, in the **Arrange** group, select **Group > Group**.



Manage comments change tracking and mailings

Objectives: At the end of this lesson you shall be able to

- collaborate in Word
- co-edit a document
- track Changes on and off

Collaborate in Word

- 1 Select Share. on the ribbon. Or, select File > Share.

Note: If your file is not already saved to OneDrive, you'll be prompted to upload your file to OneDrive to share it.

- 2 Select who you want to share with from the drop-down, or enter a name or email address.

- 3 Add a message (optional) and select Send.

Share your document

To share a file from within Word:

- 1 Select **Share**  on the ribbon.

Or, select **File > Share**.

Note: If your file is not already saved to OneDrive, you'll be prompted to upload your file to OneDrive to share it.

- 2 Select who you want to share with from the drop-down, or enter a name or email address.

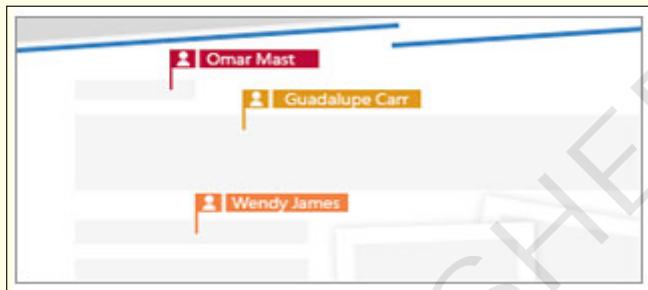
- 3 Add a message (optional) and select **Send**.



Co-edit a document

After you share your document, you can work on that file at the same time with others.

- For the best experience, work together in Word for the web and see real-time changes.
- Under **Share**, you will see the names of who else is also editing the file.
- Colored flags show you exactly where each person is working in the document.



Track and review changes

- 1 To track changes, select **Review > Track Changes**.
- 2 To review changes, place the cursor before a change and select:
 - **Accept** to keep the change, or
 - **Reject** to remove it.

Turn Track Changes on and off

On the **Review** tab, select **Track Changes**.

- When Track Changes is on, the section is highlighted. Deletions are marked with a strikethrough, and additions are marked with an underline. Different authors' changes are indicated with different colors.
- When Track Changes is off, the section is not highlighted. Word stops marking changes, but the colored underlines and strikethroughs are still in the document.

Tip: You also can add a Track Changes indicator to the status bar. Right-click the status bar and select Track Changes.

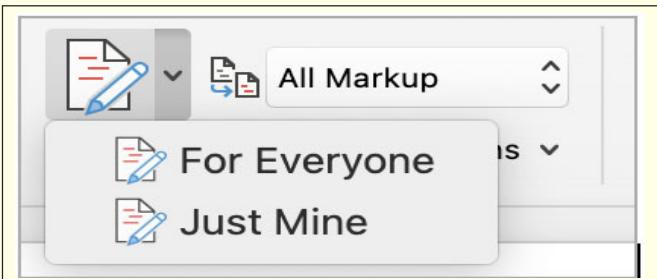
View tracked changes

Note: When a document is in Editing or Reviewing mode, you can select a tracked change to view a collaborator's full, suggested change in a card that displays. Hover over the Accept or Reject button to see a preview of what that action would do to your final document.

Choose whose changes to track

You can choose to track only your own changes or everyone's changes.

- To track only your own changes - On the **Review** tab, select **Track Changes > Just Mine**.
- To track everyone's changes - On the **Review** tab, select **Track Changes > For Everyone**.



Tip: to use a password to keep others from turning off Track Changes - On the Review tab, select Track Changes > Lock Tracking.

Choose how you would like to see the changes in the document

You can choose the type of markup you want to see.

- 1 On the Review tab, select Tracking.
- 2 Select All Markup for the drop-down list to display.
- 3 Select one of the following options.
 - **Simple Markup** displays tracked changes with a red line in the margin.
 - **All Markup** displays tracked changes with different colors of text and lines for each reviewer
 - **No Markup** hides the markup to show the document with changes incorporated
 - **Original** displays the original document without tracked changes and comments showing. However, any tracked changes or comments in the document that have not been accepted, rejected, or deleted remain in the document.

Choose the way track changes display

You can choose the types of revisions that display and the way they display. If you choose to show revisions as balloons, they display in the margins of the document. If you choose to display them directly within the document in line. In line revisions display all deletions with strikethroughs instead of inside balloons.

- 1 On the Review tab, select Track > Show Markup.
- 2 Select Balloons and then select the type of display you want.
 - Show Revisions in Balloons
 - Show All Revisions Inline
 - Show Only Formatting in Balloons

Note: If you want to see tracked changes in balloons, you must be in Print Layout view or Web Layout view.

Display changes by type of edit

- 1 On the Review tab, select Track Changes > Show Markup.
- 2 Select the type of edit.
 - Insertions and Deletions
 - Formatting

The check mark next to the item indicates that it's selected.

Note: Even if you hide a type of markup by clearing it on the Show Markup menu, the markup automatically displays each time the document is opened by you or a reviewer.

Display changes by reviewer(s)

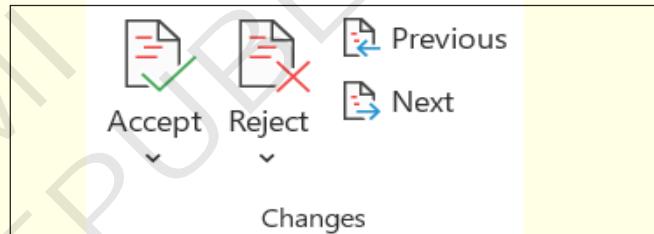
- 1 In Review > Tracking, select Show Markup.
- 2 Select Specific People.
- 3 Select to clear all check boxes except the ones next to the names of the reviewers whose changes you want to show.

Note: To select or clear all check boxes for all reviewers in the list, select All Reviewers.

Navigate tracked changes in Changes menu section

You can navigate from one tracked change to another.

- 1 In Review > Changes, select Next to view the next tracked change.
- 2 In Review > Changes, select Previous to view the previous tracked change.



Accept or Reject tracked changes

There are multiple ways to address tracked changes. The most proficient method is to identify the commentor and respond to the tracked change is through the card display when you click the marked change. The card displays the commentor name and the option to accept or reject the suggestion.

Accept or reject tracked changes in sequence using the main menu

You can resolve each tracked change from the beginning of the document to the end of the document in sequence.

- 1 Select Review > Changes > Next.
- 2 Select Accept or Reject. The next sequential tracked change is highlighted for your review and action.

Note: You can also accept or reject individual tracked changes using the main menus' drop-down menus. This method is often used as a methodical approach to reviewing tracked changes.

Accept changes using the main menu

- 1 In Review > Changes, select Accept.
- 2 Select one of the options.
 - Accept and Move to Next
 - Accept This Change

- Accept All Changes
- Accept All Changes and Stop Tracking

Reject changes using the main menu

1 In Review > Changes, select > Reject.

2 Select one of the options.

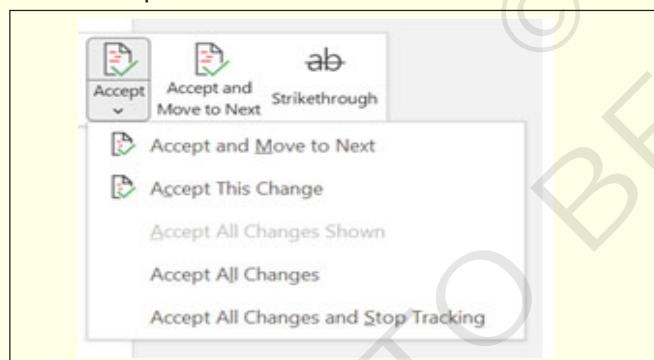
- Reject and Move to Next
- Reject This Change
- Reject All Changes
- Reject All Changes and Stop Tracking

Accept or reject changes within a document with a right click

Working within a document, you can right-click on a tracked change to accept or reject the change. Using the right-click method displays a banner menu option and a drop-down menu option.

Accept changes in-line with the right-click banner option

- 1 Right-click a tracked change in the document to review a single suggestion.
- 2 On the banner, select one of the following.
 - Accept > Accept and Move to Next
 - Accept > Accept This Change
 - Accept > Accept All Changes
 - Accept > Accept All Changes and Stop Tracking
 - Accept and Move to Next



Accept changes in-line with the right-click drop-down menu option

- 1 Right-click a tracked change in the document to review a single suggestion.
- 2 In the drop-down menu, select one of the following.
 - Accept Insertion (or Accept Deletion)
 - Reject Insertion (or Reject Deletion)

Accept or reject all tracked changes at once on the main menu

Accept all tracked changes

1 In Review > Changes, select Accept.

2 Select one of the options.

- Accept All Changes Shown
- Accept All Changes
- Accept All Changes and Stop Tracking

Reject all tracked changes

1 In Review > Changes, select Reject.

2 Select one of the options.

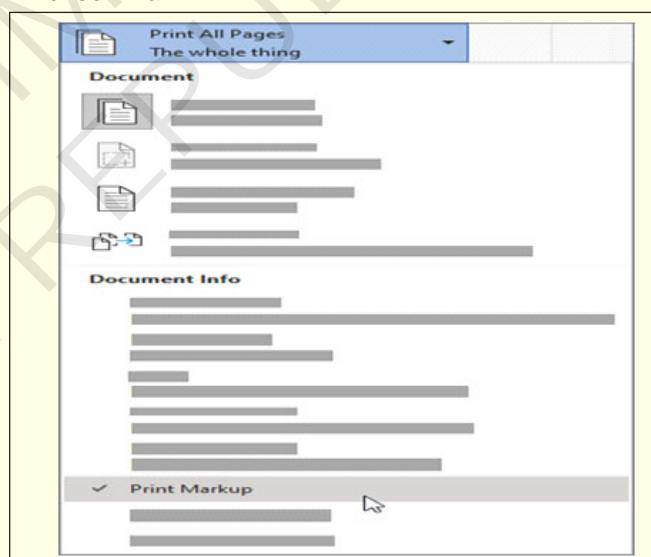
- Reject All Changes Shown
- Reject All Changes
- Reject All Changes and Stop Tracking

Note: When Review>Tracking>Show Markup>Specific People>All Reviewers is selected, the Accept All Changes Shown and Reject All Changes Shown options do not display.

Hide tracked changes and comments when printing

Hiding changes doesn't remove them from the document. You must use the Accept and Reject commands in the Changes group to remove markup from your document.

- 1 Go to File > Print > Settings > Print All Pages.
- 2 Under Document Info, select Print Markup to clear the check mark.



View all your changes in a summary list on the Reviewing Pane

- 1 In Review > Tracking, select Reviewing Pane.
- 2 Choose if you want to see the Reviewing Pane next to or below your document.
 - Select Reviewing Pane Vertical to see a list of all changes next to your document.
 - Select Reviewing Pane Horizontal to see a list of all changes below your document.

You can use the Reviewing Pane as a handy tool for to check that all tracked changes have been removed from your document so that they show up to others who might view your document. The summary section at the top of the Reviewing Pane displays the exact number of visible tracked changes and comments that remain in your document.

The Reviewing Pane also allows you to review long comments that don't fit within a comment bubble.

Note: The Reviewing Pane, unlike the document or the comment balloons, is not the best tool for making changes to your document. Instead of deleting text or comments or making other changes in the Reviewing Pane, make all editorial changes in the document. The changes will then be visible in the Reviewing Pane.

Important: Comments are no longer part of the Track Changes function.

NOT TO BE REPUBLISHED
© NIMI

Perform mail merge

Objectives: At the end of this lesson you shall be able to

- insert objects
- mail merge
- templates.

Working with object, macro, mail merge, templates and other tools

Object:

Depending on the version of Word or Outlook you're using, you can insert a variety of objects (such as PDF files, Excel charts or worksheets, or PowerPoint presentations) into a Word document or an email message by linking or embedding them. To insert an object, click Object on the Insert tab.

To create a new file that is inserted into your Word document or email message:

- 1 In the Object dialog box, click the Create New tab, and then select an option from the Object type list.

The default program for that file type opens, where you can enter any text or data you want. When you close the program, any added content or changes appear in your Word document or email message.

- 2 If you want the new file to appear as a clickable icon, rather than the first page of your new file, select Display as icon. If this check box is selected, you can choose a different icon by clicking Change Icon.

Link or embed an existing file

To link or embed an object that's already been created:

- 1 In the Object dialog box, select the Create from File tab, and then click Browse to find the file you want to insert.
- 2 To link to the source file, rather than embedding it into your Word document or email message, select Link to file.
- 3 If you want the inserted file to appear as a clickable icon, rather than the first page of the file, select Display as icon. If this check box is selected, you can choose a different icon by clicking Change Icon.

Mail Merge:

Do you have a long list of names and addresses that you need to send letters to? The Mail Merge process combines a Word document with a data source to quickly create letters that feel personal.

Mail Merge Definitions

Starting Document: (Main Document) A document that contains the information that is the same for each merged document. The starting document contains the field names for the variable information, like the names and addresses that will be inserted.

Data Source or Recipients List: A file that contains the information to be inserted into the main document during a mail merge. For example, it has records containing the names and addresses of the people a mail merge letter is sent to. Excel spreadsheets, Access databases, or Word document tables are good examples of data sources.

Field: A data category that stores a specific piece of information. For example, the field "LastName" would only contain people's last names.

Record: A record is an entire set of data fields that relate to a single thing or person. For example, a single record would include a person's first and last names, address, phone number, and date of birth.

Merge Field: A merge field is where you want to insert the information from a data source into a main document. Merge fields appear with chevrons (" ") around them. An example would be: Dear "FirstName".

Address Block: A group of merge fields that make up an address in a mail merge document. For example, a single address is made up of a name, street address, city, state, and zip code. Word can automatically insert all the appropriate address fields at once, so you don't have to insert the five or six merge fields yourself.

Greeting Line: A group of merge fields that make up the greeting line of a mail merge document, such as "Dear Mr. McDonald". Word can automatically insert all the appropriate greeting text, title, and name fields at once, so you don't have to insert the text and required merge fields yourself.

Header Row: Data source information is stored in a table. The first row of the table is the header row and contains the field names for the data source. For example, FirstName, LastName, and Address are header rows.

Set up and Choose Document Type

To begin the mail merge process, you first need to choose what sort of document you want to create.

1. Click the Mailings tab.
2. Click the Start Mail Merge button.
3. Select Step-by-Step Mail Merge Wizard.

The Mail Merge pane appears on the right, ready to walk you through the mail merge.

4. Select a type of document to create.
5. Click Next: Starting document.

The Mail Merge wizard advances to the next step.

Select a Document

This next step is to select a starting document.

1 Select a starting document.

You can use the current document as the basis for the mail merge, or you can select a template or existing document instead.

2 Click Next: Select recipients.

The Mail Merge wizard moves on to step 3.

Select Recipients

Now, you will need to choose where you'll get your list of addresses from. This example uses an existing list from a database, but you can also select Outlook contacts or manually create your own list.

1 Select Use an existing list.

2 Click Browse.

3 Select your data file.

A data file of mail merge recipients can be in a database file, an Excel spreadsheet, another Word document, or other types of data files.

4 Click Open.

The Mail Merge Recipients dialog box displays the addresses that will be used. If you're using an Excel spreadsheet as a data source, you may also be prompted to select a worksheet containing the addresses.

If there's an address you don't want to use, you can uncheck it.

5 Make sure the right recipients are selected and click OK.

6 Click Next: Write your letter.

The Mail Merge wizard moves on to step 4.

Write Your Letter

After the main document is set and the recipient list is connected and edited, you are ready to insert the merge fields in the document. The merge fields are placeholders in the document for unique information from the recipients list.

When you put a merge field in the main document, information from that field will appear for the document that is unique to that recipient.

1 Click where you want the information.

2 Select one of the placeholder options.

You can add merge fields from the wizard, or from the Write & Insert Fields group on the ribbon:

- Address Block:** This is a combination of fields to insert the names and addresses of recipients.
- Greeting Line:** This is a combination of fields to insert the recipient's name in the greeting line.

- Insert Merge Field:** When you click this button, a list of additional merge fields you can insert appears.

Customize the placeholder.

Click OK.

(Optional) Repeat steps to add each merge field you want to include.

Click Next: Preview your letters.

The Mail Merge wizard moves on to step 5.

Preview Your Mail Merge

Sometimes, it is helpful to see what the data will look like once it has been inserted into a document, instead of only viewing the merge field names.

You can easily preview how the mail merge will appear before finishing the mail merge. This is encouraged to make sure the results appear as you want them to.

1 Use the arrow buttons in the Mail Merge pane to preview each merged document.

You can also use the arrow buttons in the Preview Results group on the ribbon.

Click the Find Recipient button in the Preview Results group or in the Mail Merge pane to search for a specific recipient.

2 Click Next: Complete the merge.

The Mail Merge wizard moves on to the final step.

Complete the Merge

Once you've added the list of recipients and filled out a document with merge fields, the last step is to finish the merge by making a separate version of the document for each recipient.

There are a couple of different ways you can finish the mail merge:

- Edit Individual Documents:** Puts the results of the mail merge in a new document. You are free to edit the results of the mail merge and save and print them, just like any other document.
- Print Documents:** Merges records and sends them directly to the printer.

1 Select the option you want to use to finish the mail merge.

You can also click the Finish & Merge button on the ribbon and select a merge option there.

You're also given the option to choose which records to merge. You can merge all the records in the list, only the currently displayed record, or specify a range.

2 Select the records you want to merge.

3 Click OK.

Word merges the main document and the information from the data source into a new Word document, or merges it and sends it to the printer, based on the option you chose.

Templates

How to create a Template in Word document

A template is a pre-created document that includes some already created specific formatting options, such as macros, headers & footers, custom dictionaries, layouts, images, and AutoText entries.

A template helps you save time when creating a document with similar content and structure.

Create a template in Word document

Follow the below mentioned easiest steps to create a Template in Word document -

Step 1: Open the Word document.

Step 2: Go to the File tab on the Ribbon and click on the New option.

Step 3: Click on the Blank document.

Step 4: Add the desirable content that you use further as a template.

Save Microsoft Word Template

Once you create the template in Microsoft Word, you can save it for your further use.

In Microsoft Word, you can save Microsoft Word Template as a .dotx, .dot, or .dotm file format.

To save the template in Microsoft Word, follow the below instruction -

- 1 Open the Word document in which you create a template.
- 2 Go to the File tab on the Ribbon. A list of options will appear on the screen in which click on the Save As.
- 3 Browse the location where you want to save a file. Click on the drop-down associate with the Save as type. Select Word Template (*.dotx) and click on the Save button at the bottom of the screen.

Now, you can see that your Word template is saved in the Word document.

Edit the Word Template

Once you create the Template in Microsoft Word, you can also edit it based on your requirement and then save it for upcoming, similar Word documents.

There are the following easiest steps to edit the Word Template -

Step 1: Go to the File tab at the top left corner of the Word document and click on the Open button.

Step 2: An Open dialog box will appear on the screen. Browse the location where you save an earlier created template and click on the Open button at the bottom of the screen.

Step 3: Do the changes that you want to add to your template.

Step 4: Once you made all your desired changes in your template, you can Save your template and close it.

Use Microsoft Word Template

Microsoft Word also provides in-built Word templates that you can use on your Word document.

There are the following steps to use Microsoft Word template -

Step 1: Open the new Word document.

Step 2: Go to the File tab on the Ribbon. A File menu options will appear on the screen. Click on the New.

Step 3: An Available Templates window appears. Click on the Personal.

Step 4: A list of Word templates appears on the screen. Select and double-click on the template that you want to apply.

Now, you can see that the selected template is added to the Word document.

Open files in MS Excel

Objectives: At the end of this lesson you shall be able to

- introduction to excel
- cell Reference & linking sheet
- set or clear a print area on a worksheet.

It is a spreadsheet program developed by Microsoft. Excel organizes data in columns and rows and allows you to do mathematical functions. It runs on Windows, macOS, Android and iOS.

The first version was released in 1985 and has gone through several changes over the years. However, the main functionality mostly remains the same.

Excel is typically used for:

- | | |
|----------------------|------------------------|
| • Analysis | • Data entry |
| • Data management | • Accounting |
| • Budgeting | • Data analysis |
| • Visuals and graphs | • Programming |
| • Financial modeling | • And much, much more! |

Why Use Excel?

- It is the most popular spreadsheet program in the world
- It is easy to learn and to get started.
- The skill ceiling is high, which means that you can do more advanced things as you become better
- It can be used with both work and in everyday life, such as to create a family budget
- It has a huge community support
- It is continuously supported by Microsoft
- Templates and frameworks can be reused by yourself and others, lowering creation costs

Create a cell reference to another worksheet: Click the cell in which you want to enter the formula., type = (equal sign) and the formula you want to use. Click the tab for the worksheet to be referenced. Select the cell or range of cells to be referenced.

1. Click the cell in which you want to enter the formula.
2. In the formula bar  , type = (equal sign).
3. Do one of the following:
 - **Reference one or more cells:** To create a reference, select a cell or range of cells on the same worksheet.
 - You can drag the border of the cell selection to move the selection, or drag the corner of the border to expand the selection.
 - **Reference a defined name:** To create a reference to a defined name, do one of the following:
 - Type the name.

- Press F3, select the name in the Paste name box, and then click OK.

Note: If there is no square corner on a color-coded border, the reference is to a named range.

- 4 Do one of the following:

- If you are creating a reference in a single cell, press Enter.
- If you are creating a reference in an array formula (such A1:G4), press Ctrl+Shift+Enter.

The reference can be a single cell or a range of cells, and the array formula can be one that calculates single or multiple results.

Set or clear a print area on a worksheet: If you print a specific selection on a worksheet frequently, you can define a print area that includes just that selection. A print area is one or more ranges of cells that you designate to print when you don't want to print the entire worksheet. When you print a worksheet after defining a print area, only the print area is printed. You can add cells to expand the print area as needed, and you can clear the print area to print the entire worksheet.

A worksheet can have multiple print areas. Each print area will print as a separate page.

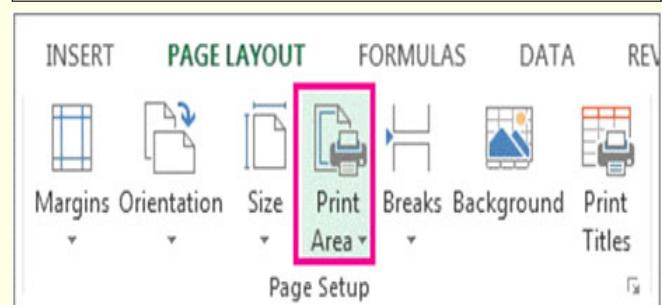
Set one or more print areas

- 1 On the worksheet, select the cells that you want to define as the print area.

Tip: To set multiple print areas, hold down the Ctrl key and click the areas you want to print. Each print area prints on its own page.

- 2 On the Page Layout tab, in the Page Setup group, click Print Area, and then click Set Print Area.

Note: The print area that you set is saved when you save the workbook.



Manipulate data

Objectives: At the end of this lesson you shall be able to

- manipulate data
 - cell range.
-

Data analysis is a challenging task, especially if you don't have the data manipulation skills. In this article, we will discuss some of the most common data manipulation techniques that can be used in Excel for data analysis.

The power of these techniques will be demonstrated by using some real-life examples. The 9 common data manipulations techniques discussed are:

- 1 Filtering
- 2 Sorting
- 3 Grouping
- 4 Pivoting
- 5 Transposing
- 6 Changing Data Types
- 7 Adding Columns and Rows
- 8 Naming Columns or Rows
- 9 Inserting Columns or Rows.

Each of these techniques will provide you with a better understanding of your data and how it works - from getting your head around different types of visualization to exploring outliers. These simple tricks will not only improve your efficiency, but also make it easier for people who don't know Excel as well to understand what you're doing.

Changing Data Types

Once you select this command, the "Select Data Type" window will appear. There are three general categories of data: Text Data

only; Numeric Data only; and Mixed Data (text and numeric).

You can further refine your selection by selecting one or more options from each of these categories: Text; Numeric; Date & Time; Logical; and Object Linking & Embedding

Adding Columns and Rows: Adding columns or rows to your data is a great way to make your work more efficient. For instance, if you were working with a table of data on different subjects and wanted to look at their answers in relation to each other, it would be more convenient for you (and the people you're sharing the data with) if you had both answers in one column.

Naming Columns and Rows: To name columns and rows in Excel, first select the cells that need naming. Then go to Data > Data Tools > Name Columns and Rows and type the name of the first cell into the first dialogue box. Continue typing or clicking until all of your cells are named.

Naming Columns or Rows

Every column and row in a spreadsheet has a default name, but these names can be changed. This is helpful when you're summarizing data and want to apply the same column or row title consistently.

To rename a column, right-click on any cell in that column and select "column name." Type in the new name and press enter. To rename a row, right-click on any cell in that row and select "Row Labels." Type in the new name and press enter.

Inserting Columns or Rows

One of the simplest data manipulation techniques in Excel is inserting columns or rows.

This technique lets you analyze your data with more clarity and precision by adding more columns or rows to your spreadsheet. It can be used to show different aspects of your data, such as different years, regions, products etc.

Two examples where this technique can be used are:

- 1 You want to compare different years for a specific region.
- 2 You want to look at how a product performs across different regions.

Conclusion

Data manipulation can be a time-consuming and complicated task. But the right technique can help you save a lot of time and avoid making mistakes. The 9 basic techniques in this article can help you navigate data manipulation in Excel.

What is a Cell Range

A cell range in an Excel file is a collection of selected cells. This range is usually symmetrical (square), but can exist of separate cells just the same. A cell range can be referred to in a formula as well.

In a spreadsheet, a cell range is defined by the reference of the upper left cell (minimum value) of the range and the reference of the lower right cell (maximum value) of the range. Eventually separate cells can be added to this selection, then the range is called an irregular cell range. In Excel, the minimum and maximum value are included. That's different from a mathematical range, in which it is a collection of values between a maximum and a minimum value.

An asymmetrical cell range can appear as below. The notation for this range is (A1:C6); from upper left cell A1 to bottom right cell C6.

	A	B	C	D
1				
2				
3				
4				
5				
6				
7				
8				

A cell range can be used inside a formula, for example to calculate the sum of the values within the selected cells. The notation for the sum of all values in cell range (A1:C6) is =SUM(A1:C6).

E4	A	B	C	D	E	F	G	H
	3	5	7					
1	4	3	6					
2	8	1	8					
3	9	1	7		88			
4	2	8	4					
5	4	3	5					
6								
7								

Create and format tables

Objectives: At the end of this lesson you shall be able to

- create and format table
 - sort & Filter
 - convert a table to text.
-

Create and format table

Formatting your range as a table tells Excel that those rows and columns are all related, and that there are headers in the first row. And by doing this, your range now has meaning. Excel understands it better. And with that, lots of additional benefits are born.

Format table

Table formatting objects. A table may contain a caption, row groups, and column groups. A row group contains rows, while a column group contains columns. Rows and columns contain cells. Tables are rendered as layers in a specified order from the bottom up: table, column groups, columns, row groups, rows, and cells.

Sorting

Sorting is another technique of data analysis and is used to rearrange the order of your data. It's an easy way of exploring and understanding your data.

For example, let's say you had a list of 5 different numbers:

1, 10, 2, 3, 4

If we wanted to sort this list in ascending order (from lowest to highest), we would click on the column heading for this list and then select "Sort Ascending". This will arrange the list like this:

1, 2, 3, 4, 10.

By sorting the numbers in ascending order we can see that they are increasing in size. If we wanted to change the sort order to descending (highest to lowest) we would click on Column A and select "Sort Descending" like so:

4, 3, 2, 10, 1.

Again by sorting in descending order from highest number to lowest number we can see that they're decreasing in size.

Filtering

Filtering is a process of sorting data by a certain criteria. It's an effective way to identify subsets of data from the larger dataset.

In the following example, we have a dataset with monthly sales data from 2012-2015. Filtering is useful if you want to see the total sales for the year only, or if you want to know how many months had positive growth.

First, identify which column contains the filter criteria that will be used to filter your data. In this case, I have created a new column called "Sales Growth." Next, highlight your "Sales Growth" column and select "Filter" from the Data menu on the toolbar. This will open a dialogue box where you can input your filtering criteria. In this example, I am using ">0%" as my filter criterion to calculate total sales for years with positive revenue growth > 0%.

Convert a table to text

- 1 Select the rows or table you want to convert to text.
- 2 On the Layout tab, in the Data section, click Convert to Text.
- 3 In the Convert to Text box, under Separate text with, click the separator character you want to use in place of the column boundaries. ...
- 4 Click OK.

Functions and formulas in MS-Excel 2010

Objectives: At the end of this lesson you shall be able to

- features & Functions of Microsoft Excel
- formulas and Functions
- move Around in Excel 2010
- conditional Formatting
- link Excel Spreadsheet Data

Formulas in MS Excel

formula, worksheet will be just simple tabular representation of data. A formula consists of special code, which is entered into a cell. It performs some calculations and returns a result, which is displayed in the cell.

Formulas use a variety of operators and worksheet functions to work with values and text. The values and text used in formulas can be located in other cells, which makes changing data easy and gives worksheets their dynamic nature. For example, it can quickly change the data in a worksheet and formulas works.

Elements of Formulas

A formula can consist of any of these elements ?

- Mathematical operators, such as +(for addition) and *(for multiplication)

Example -

=A1+A2 Adds the values in cells A1 and A2.

- Values or text

Example -

=200*0.5 Multiplies 200 times 0.5. This formula uses only values, and it always returns the same result as 100.

- Cell references (including named cells and ranges)

Example -

=A1=C12 Compares cell A1 with cell C12. If the cells are identical, the formula returns TRUE; otherwise, it returns FALSE.

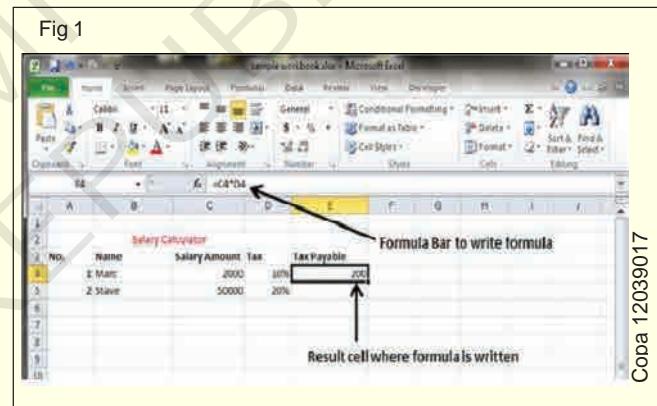
- Worksheet functions (such as SUM or AVERAGE)

Example -

=SUM(A1:A12) Adds the values in the range A1:A12.

Creating Formula

For creating a formula need to type in the Formula Bar. Formula begins with '=' sign. When building formulas manually, and can either type in the cell addresses or can point to them in the worksheet. Using the Pointing method to supply the cell addresses for formulas is often easier and more powerful method of formula building. When using built-in functions, to click the cell or drag through the cell range that want to use when defining the function's arguments in the Function Arguments dialog box as shown in Fig 1.



As soon as complete a formula entry, Excel calculates the result, which is then displayed inside the cell within the worksheet (the contents of the formula, however, continue to be visible on the Formula bar anytime the cell is active). If you make an error in the formula that prevents Excel from being able to calculate the formula at all, Excel displays an Alert dialog box suggesting how to fix the problem.

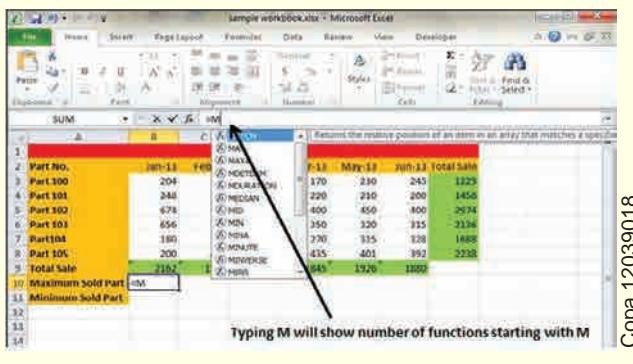
Functions in Formula

Many formulas are created using available worksheet functions. These functions enable to greatly enhance the power of the formulas and perform calculations that are difficult if use only the operators. For example, one can use the LOG or SIN function to calculate the Logarithm or Sin ratio. And cannot do this complicated calculation by using the mathematical operators alone.

Using Functions

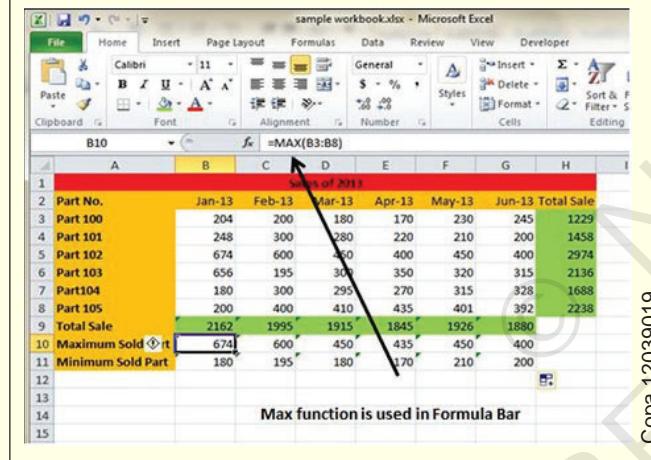
When type = sign and then type any alphabet the searched functions will show Fig 2.

Fig 2



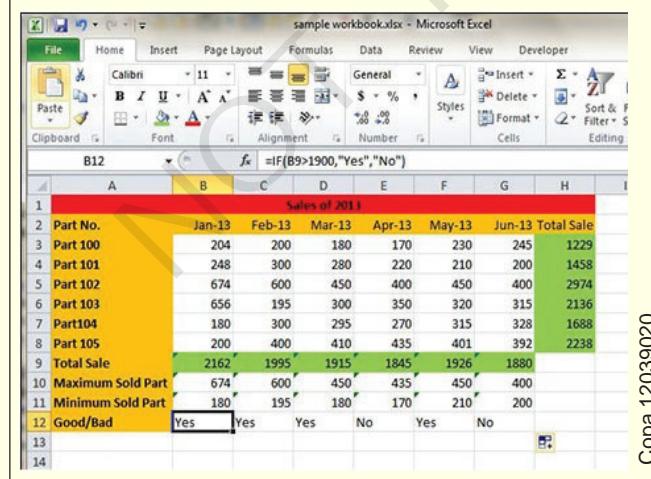
Suppose need to determine the largest value in a range. A formula can't tell the answer without using a function. We will use formula that uses the MAX function to return the largest value in the range B3:B8 as **=MAX(A1:D100)** as shown in Fig 3.

Fig 3



Another example of functions. Suppose to find if the cell of month is greater than 1900 then we can give Bonus to Sales representative. The we can achieve it with writing formula with IF functions as **=IF(B9>1900,"Yes","No")** as shown in Fig 4.

Fig 4



Function Arguments

In the above examples, notice that all the functions used parentheses. The information inside the parentheses is the list of arguments.

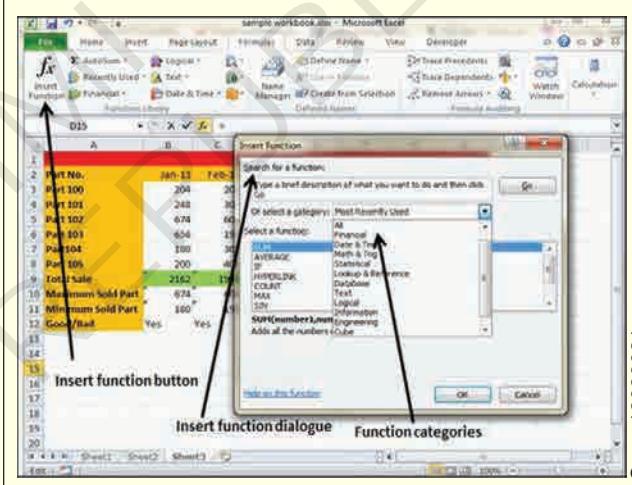
Functions vary in how they use arguments. Depending on what it has to do, a function may use.

- **No arguments** - Examples ? Now(), Date(), etc.
- **One argument** - UPPER(), LOWER(), etc.
- **A fixed number of arguments** - IF(), MAX(), MIN(), AVERAGE(), etc.
- **Infinite number of arguments**
- **Optional arguments**

Built In Functions

MS Excel has many built in functions, which we can use in our formula. To see all the functions by category, choose Formulas Tab " Insert Function as shown in Fig 5. Then Insert function Dialog appears from which we can choose the function.

Fig 5



Functions by Categories

Let us see some of the built in functions in MS Excel.

Text Functions

LOWER: Converts all characters in a supplied text string to lower case

UPPER : Converts all characters in a supplied text string to upper case

TRIM : Removes duplicate spaces, and spaces at the start and end of a text string

CONCATENATE : Joins together two or more text strings.
Fig 4

LEFT : Returns a specified number of characters from the start of a supplied text string.

MID : Returns a specified number of characters from the middle of a supplied text string

RIGHT : Returns a specified number of characters from the end of a supplied text string.

LEN: Returns the length of a supplied text string

FIND: Returns the position of a supplied character or text string from within a supplied text string (case-sensitive).

Date & Time

DATE: Returns a date, from a user-supplied year, month and day.

TIME: Returns a time, from a user-supplied hour, minute and second.

DATEVALUE: Converts a text string showing a date, to an integer that represents the date in Excel's date-time code.

TIMEVALUE: Converts a text string showing a time, to a decimal that represents the time in Excel.

NOW: Returns the current date & time.

TODAY: Returns today's date.

Statistical

MAX: Returns the largest value from a list of supplied numbers.

MIN: Returns the smallest value from a list of supplied numbers.

AVERAGE: Returns the Average of a list of supplied numbers.

COUNT: Returns the number of numerical values in a supplied set of cells or values.

COUNTIF: Returns the number of cells (of a supplied range), that satisfies a given criteria.

SUM: Returns the sum of a supplied list of numbers

Logical

AND: Tests a number of user-defined conditions and returns TRUE if ALL of the conditions evaluate to TRUE, or FALSE otherwise

OR: Tests a number of user-defined conditions and returns TRUE if ANY of the conditions evaluate to TRUE, or FALSE otherwise.

NOT: Returns a logical value that is the opposite of a user supplied logical value or expression i.e. returns FALSE if the supplied argument is TRUE and returns TRUE if the supplied argument is FAL

Math & Trig

ABS: Returns the absolute value (i.e. the modulus) of a supplied number.

SIGN: Returns the sign (+1, -1 or 0) of a supplied number.

SQRT: Returns the positive square root of a given number.

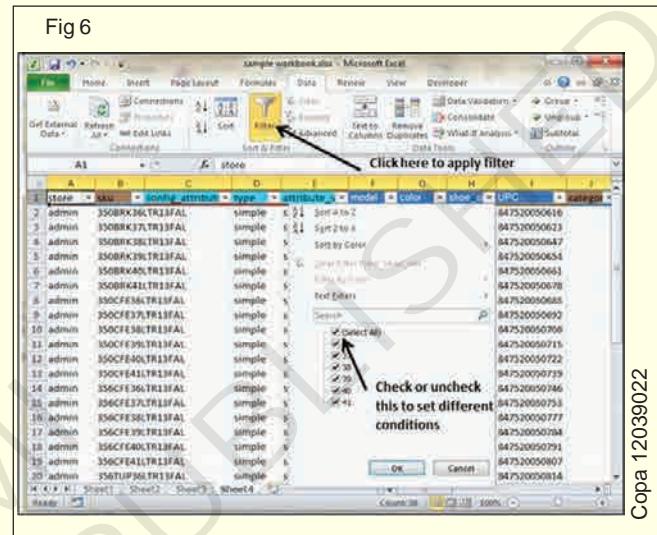
MOD: Returns the remainder from a division between two supplied numbers.

Filters in MS Excel

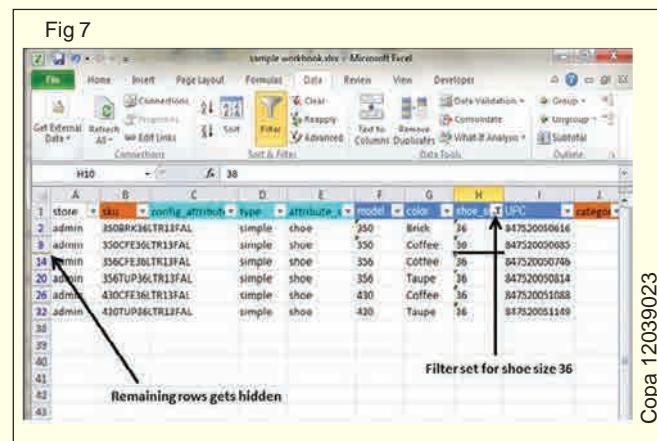
Filtering data in MS Excel refers to displaying only the rows that meet certain conditions. (The other rows gets hidden.)

Using the store data, if user interested in seeing data where Shoe Size is 36, then set filter to do this. Follow the below mentioned steps to do this.

- Place a cursor on the Header Row.
- Choose **Data Tab** ➤ **Filter** to set filter as shown in Fig 6.



- Click the drop-down arrow in the Area Row Header and remove the check mark from Select All, which unselects everything.
- Then select the check mark for Size 36 which will filter the data and displays data of Shoe Size 36 as shown in fig-23.
- Some of the row numbers are missing; these rows contain the filtered (hidden) data.
- There is drop-down arrow in the Area column now shows a different graphic - an icon that indicates the column is filtered. (Fig 7)



Copa 12039022

Copa 12039023

Using Multiple Filters

Filtering of records by multiple conditions i.e. by multiple column values. Suppose after size 36 is filtered, need to have the filter where color is equal to Coffee. After setting filter for Shoe Size, choose Color column and then set filter for color. (Fig 8)

Fig 8

Filtered Records

Filter set on Color

Copa 12039024

Sorting in MS Excel

Sorting data in MS Excel rearranges the rows based on the contents of a particular column. sort a table to put names in alphabetical order Or sort data by Amount from smallest to largest or largest to smallest.

To Sort the data follow the steps mentioned below.

- Select the Column to sort data as on Fig 9.

Fig 9

Select the column

Click Sort to sort data

Copa 12039027

- Choose Data Tab " Sort Below dialog appears.
- If the user sort data based on a selected column, Choose Continue with the selection or if the data sorting based on other columns, choose Expand Selection.
- Sort the data based on the below Conditions as on Fig 10.

Fig 10

Sort Dialogue

Column to sort by

Copa 12039026

- Values** - alphabetically or numerically.
- Cell Color** - Based on Color of Cell.
- Font Color** - Based on Font color.
- Cell Icon** - Based on Cell Icon.
- Clicking Ok will sort the data as on Fig 11.

Fig 11

Sorted Records by CustomerLast name

Copa 12039027

Sorting option is also available from the Home Tab. Choose Home Tab " Sort & Filter. You can see the same dialog to sort records as on Fig 12.

Fig 12

Sort & Filter option

Home Tab

Copa 12039028

Ranges in MS Excel

A cell is a single element in a worksheet that can hold a value, some text, or a formula. A cell is identified by its address, which consists of its column letter and row number. For example, cell B1 is the cell in the second column and the first row.

A group of cells is called a range. You designate a range address by specifying its upper-left cell address and its lower-right cell address, separated by a colon.

Example of Ranges:

- C24** - A range that consists of a single cell.
- A1:B1** - Two cells that occupy one row and two columns.
- A1:A100** - 100 cells in column A.
- A1:D4** - 16 cells (four rows by four columns).

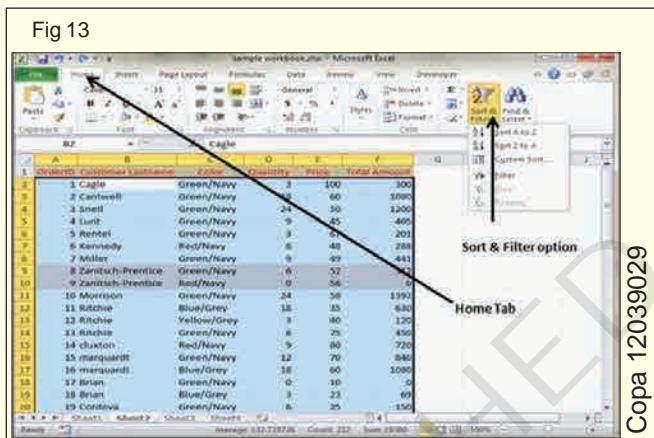
Selecting Ranges

Selecting a range in several ways ?

- Press the left mouse button and drag, highlighting the range. Then release the mouse button. If drag to the end of the screen, the worksheet will scroll.
- Press the Shift key while uses the navigation keys to select a range.
- Press F8 and then move the cell pointer with the navigation keys to highlight the range. Press F8 again to return the navigation keys to normal movement.

- Type the cell or range address into the Name box and press Enter. Excel selects the cell or range that specified. (Fig 13)

Selecting Complete Rows and Columns



When user need to select an entire row or column.and can select entire rows and columns in much the same manner as select ranges:

- Click the row or column border to select a single row or column.
- To select multiple adjacent rows or columns, click a row or column border and drag to highlight additional rows or columns.
- To select multiple (nonadjacent) rows or columns, press Ctrl while click the row or column borders. (Fig 14)

Fig 14

OrderID	Customer Lastname	Color	Quantity	Price	Total Amount
1	Cagle	Green/Navy	3	100	300
2	Cantwell	Green/Navy	18	50	1080
3	Snell	Green/Navy	24	50	1200
4	Lunt	Green/Navy	9	45	405
5	Rentel	Green/Navy	3	67	201
6	Kennedy	Red/Navy	6	48	288
7	Miller	Green/Navy	9	49	441
8	Zanitsch-Prentice	Green/Navy	6	52	312
9	Zanitsch-Prentice	Red/Navy	0	56	0
10	Morrison	Green/Heavy	24	56	1344
11	Ritchie	Blue/Grey	18	35	630
12	Ritchie	Yellow/Grey	3	40	120
13	Ritchie	Green/Navy	8	25	200
14	Claxton	Red/Navy	9	60	540
15	Marquardt	Green/Nav	12	70	840
16	Marquardt	Blue/Grey	18	60	1080
17	Brian	Green/Navy	0	10	0
18	Brian	Blue/Grey	3	25	75
19	Centofanti	Green/Navy	6	56	336

Copa 12039030

Manage Charts

Objectives: At the end of this lesson you shall be able to

- create charts
- modify charts
- format charts.

Chart

Insert a chart to illustrate and compare data.

Bar, Pie, Line, Area and Surface are some of the available types.

Create a chart (graph) that is recommended for your data, almost as fast as using the chart wizard that is no longer available.

Create a chart

- 1 Select the data for which you want to create a chart.
- 2 Click INSERT > Recommended Charts.
- 3 On the Recommended Charts tab, scroll through the list of charts that Excel recommends for your data, and click any chart to see how your data will look.
If you don't see a chart you like, click All Charts to see all the available chart types.
- 4 When you find the chart you like, click it > OK.
- 5 Use the Chart Elements, Chart Styles, and Chart Filters buttons, next to the upper-right corner of the chart to add chart elements like axis titles or data labels, customize the look of your chart, or change the data that is shown in the chart.
- 6 To access additional design and formatting features, click anywhere in the chart to add the CHART TOOLS to the ribbon, and then click the options you want on the DESIGN and FORMAT tabs

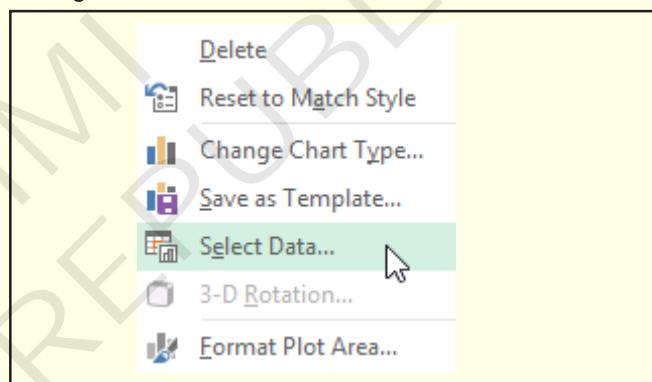
What is data series in Excel chart?

What is a Data Series in Excel? A data series is a collection of consecutive values or numbers in a column on a spreadsheet. We can create data series by plotting a graph from an initial series of data, then, we can create another series of data and highlight it along with the pre-existing data series

Add a data series to a chart on a separate chart sheet

If your chart is on a separate worksheet, dragging might not be the best way to add a new data series. In that case, you can enter the new data for the chart in the Select Data Source dialog box.

- 1 On the worksheet that contains your chart data, in the cells directly next to or below your existing source data for the chart, enter the new data series you want to add.
- 2 Click the worksheet that contains your chart.
- 3 Right-click the chart, and then choose Select Data.

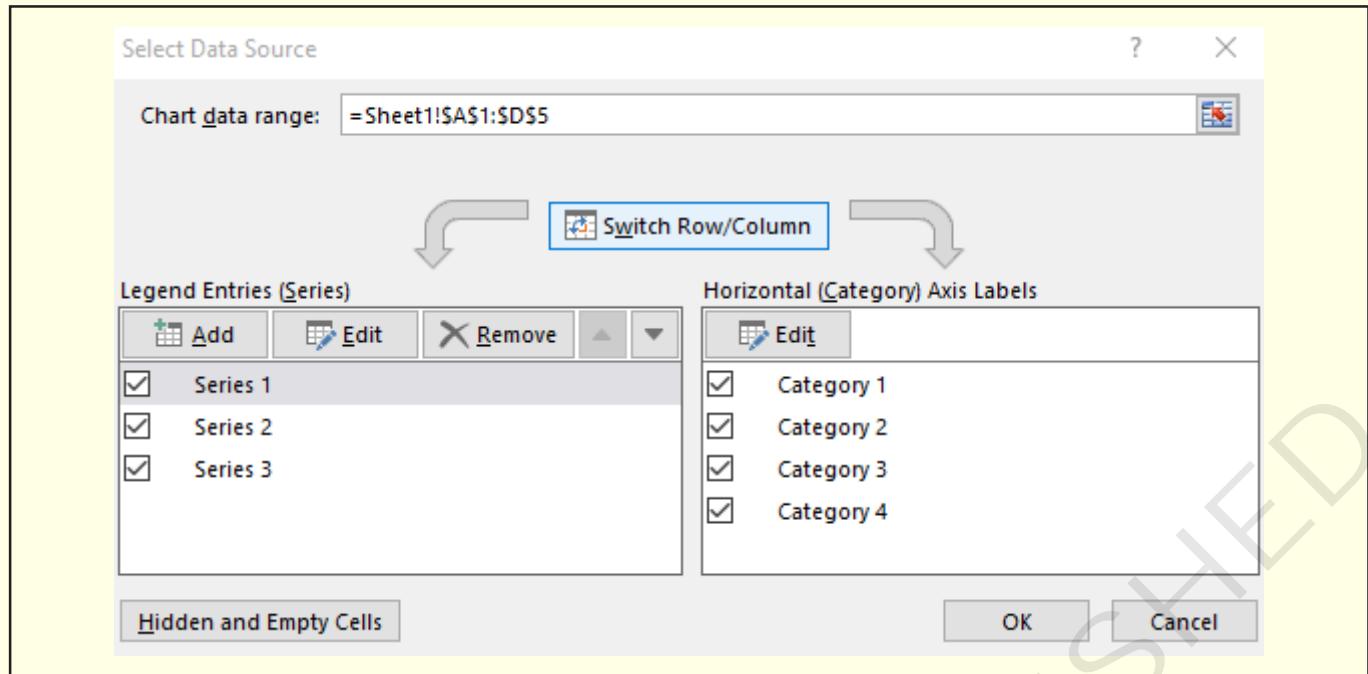


The Select Data Source dialog box appears on the worksheet that contains the source data for the chart.

- 4 Leaving the dialog box open, click in the worksheet, and then click and drag to select all the data you want to use for the chart, including the new data series.

The new data series appears under Legend Entries (Series) in the Select Data Source dialog box.

- 5 Click OK to close the dialog box and to return to the chart sheet.



Create Pivot Tables

Objectives: At the end of this lesson you shall be able to

- pivot table
- pivot table work
- group data.

Pivot table

A PivotTable is an interactive way to quickly summarize large amounts of data. You can use a PivotTable to analyze numerical data in detail, and answer unanticipated questions about your data. A PivotTable is especially designed for: Querying large amounts of data in many user-friendly ways.

Pivot table work

A Pivot Table is used to summarise, sort, reorganise, group, count, total or average data stored in a table. It allows us to transform columns into rows and rows into columns. It allows grouping by any field (column), and using advanced calculations on them.

Group data

Grouping data in a PivotTable can help you show a subset of data to analyze. For example, you may want to group an unwieldy list date and time fields in the PivotTable into quarters and months

Open files in MS PowerPoint Presentations

Objectives: At the end of this lesson you shall be able to

- introduction to MS Power Point Presentation
 - apply and edit picture effects.
-

The Microsoft PowerPoint is a simple yet powerful software program developed by Microsoft to produce more effective presentations. It is mainly a part of the Microsoft Office suite. The program comprises of slides and various such useful tools like word processing, drawing, graphing and outlining. Thus it can be used to display text, tables, charts, graphics and media in the slides with ease.

There are mainly three noteworthy features of the Microsoft PowerPoint window that you need to focus upon while learning more about PowerPoint. These features are called the Microsoft Office Button, Quick Access Toolbar and the Ribbon.

A PowerPoint slideshow (PPT) is a presentation created on software from Microsoft that allows users to add audio, visual and audio/visual features to a presentation. It is considered to be a multimedia technology and also acts as a tool for collaboration and content sharing. PowerPoint is included in Microsoft Office, making it one of the most well-known and widely used brands of presentation software.

A PowerPoint slideshow is also known as a PowerPoint presentation.

Edit pictures

- 1 Adjust the brightness, contrast, or sharpness. Select the picture. Select Picture Format and select Corrections. ...
- 2 Apply artistic effects. Select the picture. ...
- 3 Change the color. Select the picture. ...
- 4 Apply picture effects. Select the picture. ...
- 5 Add a border. Select the picture. ...
- 6 Compress the picture. Select the picture.
 - Adjust the brightness, contrast, or sharpness
 - Select the picture.
 - Select Picture Format and select Corrections.
 - Hover over the options to preview them and select the one you want.

For more info, see Change the brightness, contrast, or sharpness of a picture.

Apply artistic effects

- 1 Select the picture.
- 2 Select Picture Format and select Artistic Effects.
- 3 Hover over the options to preview them, then select the one you want.

Note: You can apply only one artistic effect at a time to a picture, so applying a different artistic effect will remove the previously applied artistic effect.

- 4 Change the color
- 5 Select the picture.
- 6 Select Picture Format and select Color.
- 7 Hover over the options to preview them, then select the one you want.

Apply picture effects

- 1 Select the picture.
- 2 Select Picture Format and select Picture Effects.
- 3 Select the one you want: Shadow, Reflection, Glow, Soft Edges, Bevel, or 3-D Rotation.

For more info, see Add or change an effect for a picture.
- 4 Add a border
- 5 Select the picture.
- 6 Select Picture Format > Picture Border, then select a border.

For more info, see Add and remove picture borders.
- 7 Remove the background

For info, see Remove a picture background.
- 8 Compress the picture
- 9 Select the picture.
- 10 Select Picture Format and select Compress Pictures.
- 11 Select the options you want, then select OK.

Insert, Format text and paragraphs

Objectives: At the end of this lesson you shall be able to

- apply text alignment.

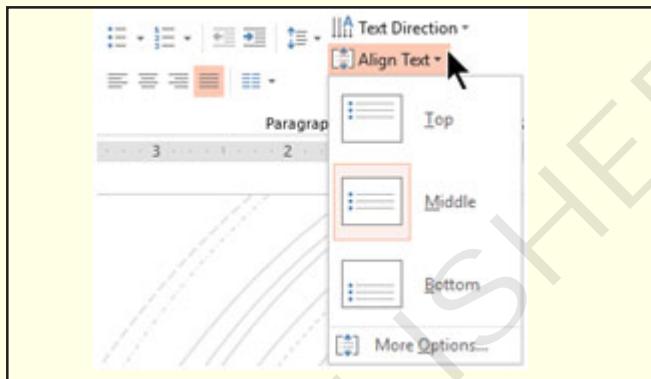
How To Change Text Alignment

- Select the required text
- Now, In the Home tab locate the Paragraph group
- There are four alignment options available at the bottom of the Paragraph group
- Select your desired alignment option with a left click.

The four alignment options available are:

- **Align Text Left:** It Aligns your text towards the left margin
- **Center:** Brings the text towards the center
- **Align Text Right:** Aligns your text towards the right margin

- **Justify:** Aligns the text towards both the left and right margins



COPA - Manage tables and bulleted text**Create tables, modify tables, modify bulleted text**

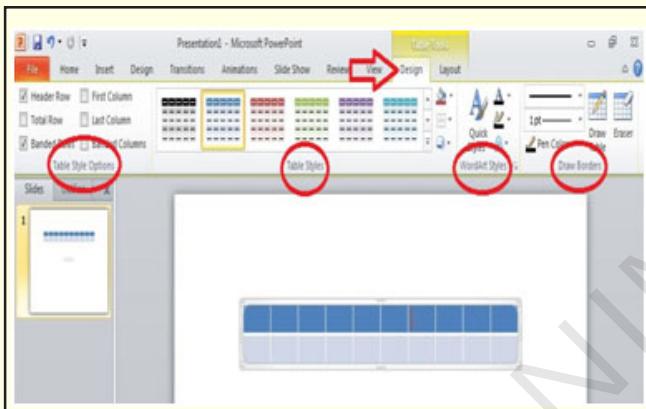
Objectives: At the end of this lesson you shall be able to

- modify or format table
- apply bulleted and numbered list.

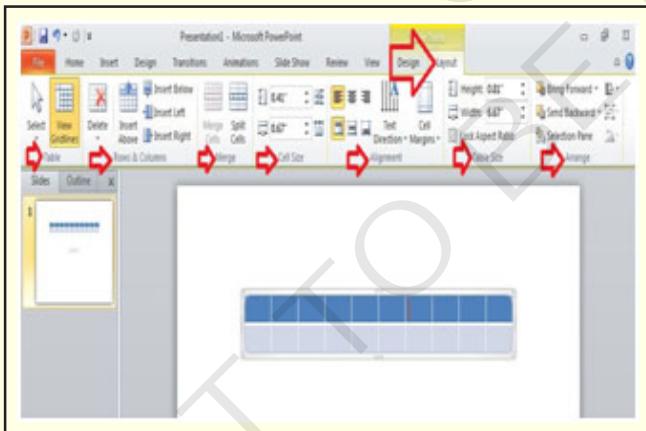
How To Modify Or Format Table

Select that particular table that you want to modify. Two new tabs called Design and Layout will now appear in the Ribbon. These tabs offer various new groups of commands to format your tables.

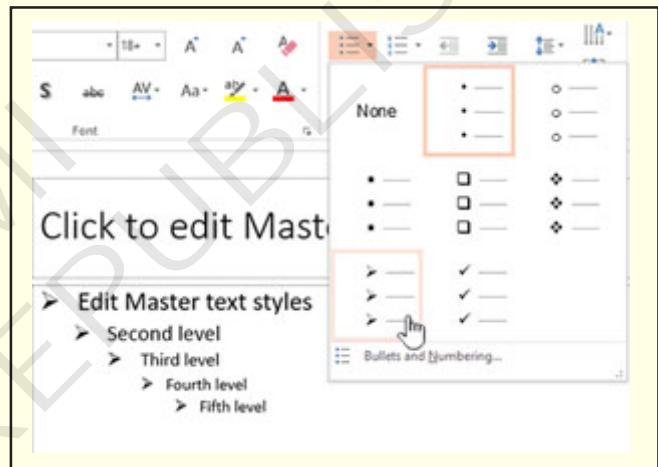
When you click on the Design tab it displays to you four new groups of commands; Table Style Options, Table Styles, Word Art Styles and the Draw Borders.



When you click on the Layout tab it will display six groups of commands; Table, Rows & Columns, Merge, Cell Size and Alignment.

**How To Create Bulleted Or Numbered Lists**

- First, select the text or the list to which you want to add your bullets or numbering
- Now, Select the Home tab and locate the Paragraph group
- Click the on Bullets or Numbering button or click on the drop-down arrow next to these buttons to see even more bullet styles and numbering formats
- Select your style that you want to apply to text



Insert illustrations, Format illustrations and text boxes

Objectives: At the end of this lesson you shall be able to

- add text to a slide
- change layout color or style
- format smart art.

Add text to a slide

You can add text to a PowerPoint slide or a slide master by inserting a text box and typing inside that box. You can then format that text by selecting the text or the entire box. You can also add text to placeholders and shapes.

Select a heading below to open it and see the detailed instructions.

Add text to a text box

- To add text to a text box that anyone can edit, in Normal view, click inside the text box, and then type or paste the text.
- To add text to a text box that is contain permanent and un-editable, in Slide Master view, click inside the text box, and then type or paste the text.

Use text boxes to place text anywhere on a slide, such as outside a text placeholder. For example, to add a caption to a picture, create a text box and position it near the picture.

Add text that is part of a shape

Shapes such as squares, circles, callout balloons, and block arrows can contain text. When you type text into a shape, the text attaches to the shape and moves and rotates with it.

- To add text that becomes part of a shape, select the shape, and then type or paste the text.

Add text that is independent of a shape

A text box is handy if you want to add text to a shape, but you don't want the text to attach to the shape. You can add a border, fill, shadow, or three-dimensional (3-D) effect to text in a text box.

- To add text that moves independently of a shape, add a text box, and then type or paste the text.

Add a text box

- a On the Insert tab, in the Text group, click Text Box.

Note: If you are using an East Asian language, from the Text Box drop-down menu, click either Horizontal or Vertical alignment.

- b Click the slide, and then drag the pointer to draw the text box.

Change the layout, color or style

Select the SmartArt. Select the SmartArt Tools > Design tab. Hover over the Layouts to preview them, and select the one you want. Select Change Colors, hover over the options to preview them, and select the one you want.

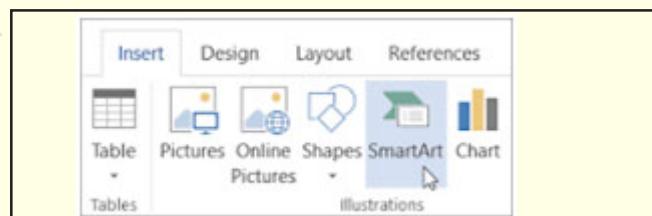
How do you Format smart art?



Format SmartArt

Click the SmartArt shape you want to format. Click the Format tab under SmartArt Tools. Use the options in the Shape Styles group to format each shape.

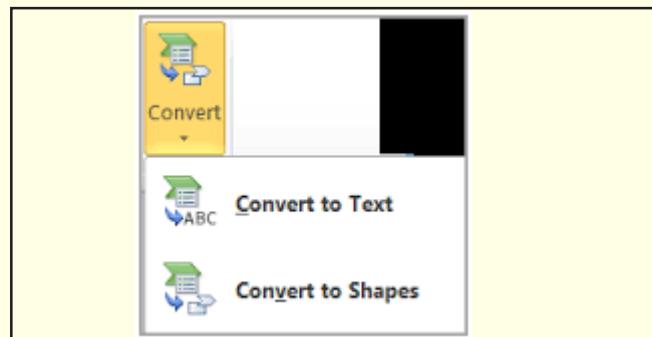
How do you insert and Format a SmartArt graphic?



Insert a SmartArt graphic and add text to it

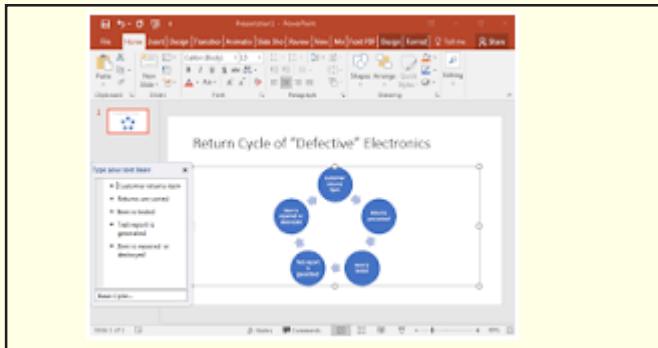
On the Insert menu, select SmartArt Graphic. On the SmartArt tab of the ribbon, in the Insert SmartArt Graphic group, select at the type of graphic you want (List, Process, etc.), and then select a layout.

How do I change the shape of Smart Art in PowerPoint?



Under SmartArt Tools, on the Design tab, in the Reset group, click Convert, and then click Convert to Shapes. If you do not see the SmartArt Tools or Design tabs, make sure that you have selected a SmartArt graphic. You might have to double-click the shape to open the Design tab.

What is use of smart shapes in PowerPoint explain with example?



SmartArt is a dynamic type of image that you will often see in PowerPoint slides. SmartArt can be used to group lists of information together, through bullet lists for example, or to show a process, through cycles.

Which are types of SmartArt graphic?

Layout types. The Choose a SmartArt Graphic gallery displays all available layouts broken into eleven different types - All, List, Process, Cycle, Hierarchy, Relationship, Matrix, Pyramid, Picture, Office.com, and Other.



Why is SmartArt important in PowerPoint?

Smart Art helps create organization, hierarchy, stages and processes. It introduces visually pleasing colours and diagrams and adds value to what the speaker is saying. There are several types of SmartArts that one can use, based on the requirements and kind of information

Audio & Video Elements

Objectives: At the end of this lesson you shall be able to

- add or delete audio in PPP
 - record Audio
 - change playback options.
-

Add or delete audio in your PowerPoint presentation

You can add audio, such as music, narration, or sound bites, to your PowerPoint presentation. To record and hear any audio, your computer must be equipped with a sound card, microphone, and speakers.

Add audio from your PC

- 1 Select Insert > Audio.
- 2 Select Audio on My PC.
- 3 In the **Insert Audio** dialog box, select the audio file you want to add.
- 4 Select Insert.

Record audio

- 1 Select Insert > Audio.
- 2 Select Record Audio.
- 3 Type in a name for your audio file, select Record, and then speak.

Note: Your device must have a microphone enabled in order to record audio.

- 4 To review your recording, select Stop and then select Play.
- 5 Select Record to re-record your clip, or select OK if you're satisfied.
- 6 To move your clip, select and drag the audio icon to where you want it on the slide.

If you're using more than one audio file per slide, we recommend putting the audio icon in the same spot on a slide to find it easily.

- 7 Select Play.

Change playback options

Select the audio icon and then select the Audio Tools Playback tab. Then select which options you'd like to use:

- To trim the audio, select Trim and then use the red and green sliders to trim the audio file accordingly.
- To fade in or fade out audio, change the number in the Fade Duration boxes.
- To adjust volume, select Volume and select the setting you prefer.
- To choose how the audio file starts, select the dropdown arrow and select an option:
 - In Click Sequence: Plays the audio file automatically with a click.
 - Automatically: Plays automatically once you advance to the slide that the audio file is on.
 - When Clicked On: Plays audio only when the icon is clicked on.
- To choose how the audio plays in your presentation, select an option:
 - Play Across Slides: Plays one audio file across all slides.
 - Loop until Stopped: Plays an audio file on loop until it's stopped manually by clicking the Play/Pause button.
 - To have the audio play continuously across all slides in the background, select Play in Background.

Delete audio

To delete an audio clip, select the audio icon on the slide and press Delete.

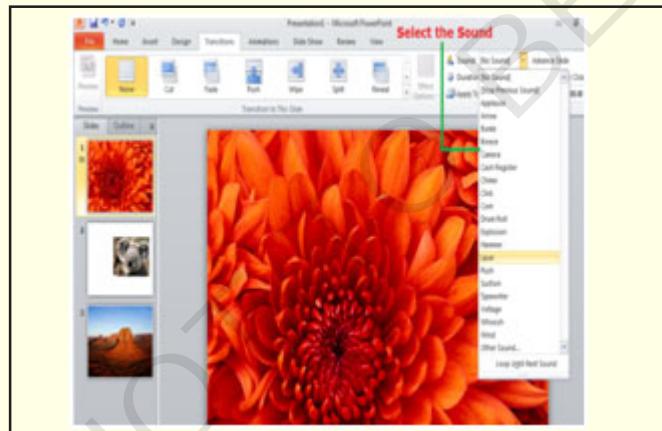
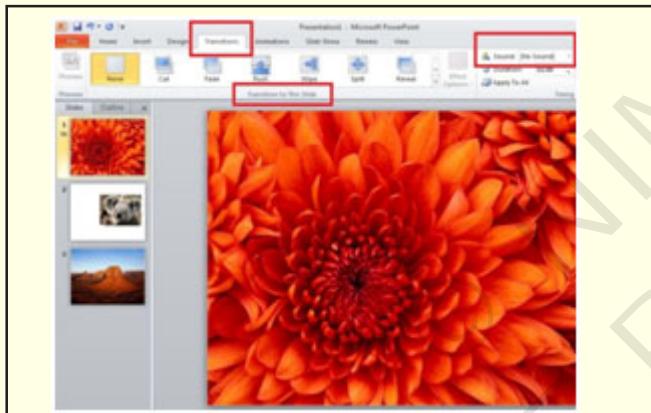
COPA - Manage Transitions and Animations**Add slide transitions & animations**

Objectives: At the end of this lesson you shall be able to

- set slide transition sound
- apply slide transition effects
- apply a custom animation effect.

How To Set Slide Transition Sound

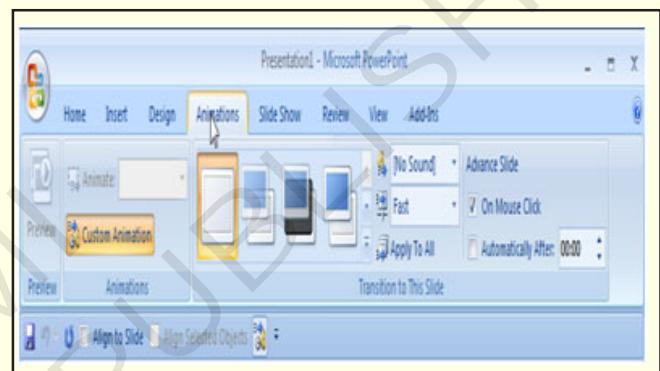
- First, Select the slide to which you want to add the transition sound
- Select the Transition tab
- In the Transition to This Slide group click on the drop-down arrow next to the Transition Sound option
- A list of several different transition sounds appears
- Select your desired transition sound
- Click on the Apply To All to apply the effect to all the slides

**How To Apply Slide Transition Effects**

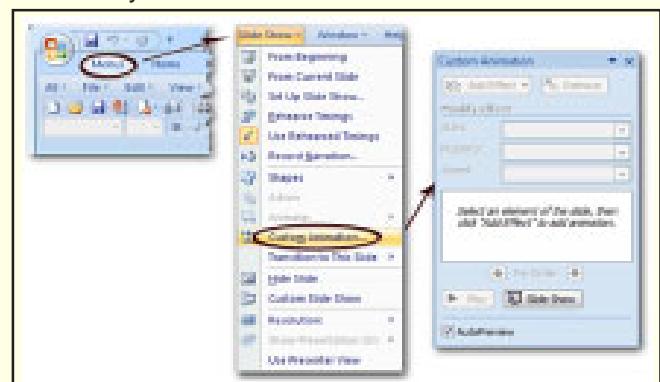
Transition effects usually appear when one slide changes into the next slide in a Slide Show.

- First, Select the slide to which you want to apply the effect
- Select the Animation tab

- Now, In the Transition to This Slide group you will see the transition effects
- Click on the drop-down arrow to see the menu of transition effects
- Select your desired transition effect
- Click on Apply To All to apply the effect to all the slides

**How To Apply A Custom Animation Effect**

- First, Select the required text or object
- Then, Select the Animation tab
- In the Animation group click on the Custom Animation
- Custom Animation task pane now appears on the right
- Click on the Add Effect, it will display four more options that are further divided into different options
- Select your desired effect



Add and manage comments

Objectives: At the end of this lesson you shall be able to

- add a command
- show or Hide command
- edit & Delete a command.

Add, change, hide, or delete comments in a presentation

Use comments when you want people to review and provide feedback on a presentation that you created, or when colleagues ask for your feedback on a presentation. A comment is a note that you can attach to a letter or word on a slide, or to an entire slide.

Add a comment

- 1 Select the object or slide you want to comment on. Select Review > New Comment. Or select New if the Comments pane is open. You can also add a comment by selecting Insert > Comment.
- 2 In the Comments pane, type your message in the box and select Post or press Ctrl+Enter.

Show or hide comments

- 1 On the View tab, click Normal.
- 2 On the Review tab, click Show Comments.

View and reply to comments

- 1 Select the comment in the Comments pane.
- 2 Select Reply to respond to a comment.

Edit comments

- 1 In the navigation pane, in Normal view, click the slide you want to comment on.
- 2 In the Comments pane, click the comment you want to edit, and then click the pencil icon to make changes.

Note: Keep in mind that it's possible for others to edit your comments. Comments in an Office document are stored in the file, so anyone with edit access to your file can edit your comment.

Delete a comment

In the Comments pane, select the comment you want to delete, select More thread actions (...) and select Delete thread.

Install, Troubleshoot, Create and Use of database in MySQL

Objectives: At the end of this lesson you shall be able to

- concept of DBMS & RDBMS
- difference of DBMS & RDBMS
- define data models
- concept of DBA
- define database Users
- define database schema.

Concepts of DBMS & RDBMS

A relational database management system (RDBMS) is a collection of programs and capabilities that enable IT teams and others to create, update, administer and otherwise interact with a relational database. RDBMSes store data in the form of tables, with most commercial relational database management systems using Structured Query Language (SQL) to access the database. However, since SQL was invented after the initial development of the relational model, it is not necessary for RDBMS use.

The RDBMS is the most popular database system among organizations across the world. It provides a dependable method of storing and retrieving large amounts of data while offering a combination of system performance and ease of implementation.

RDBMS vs. DBMS

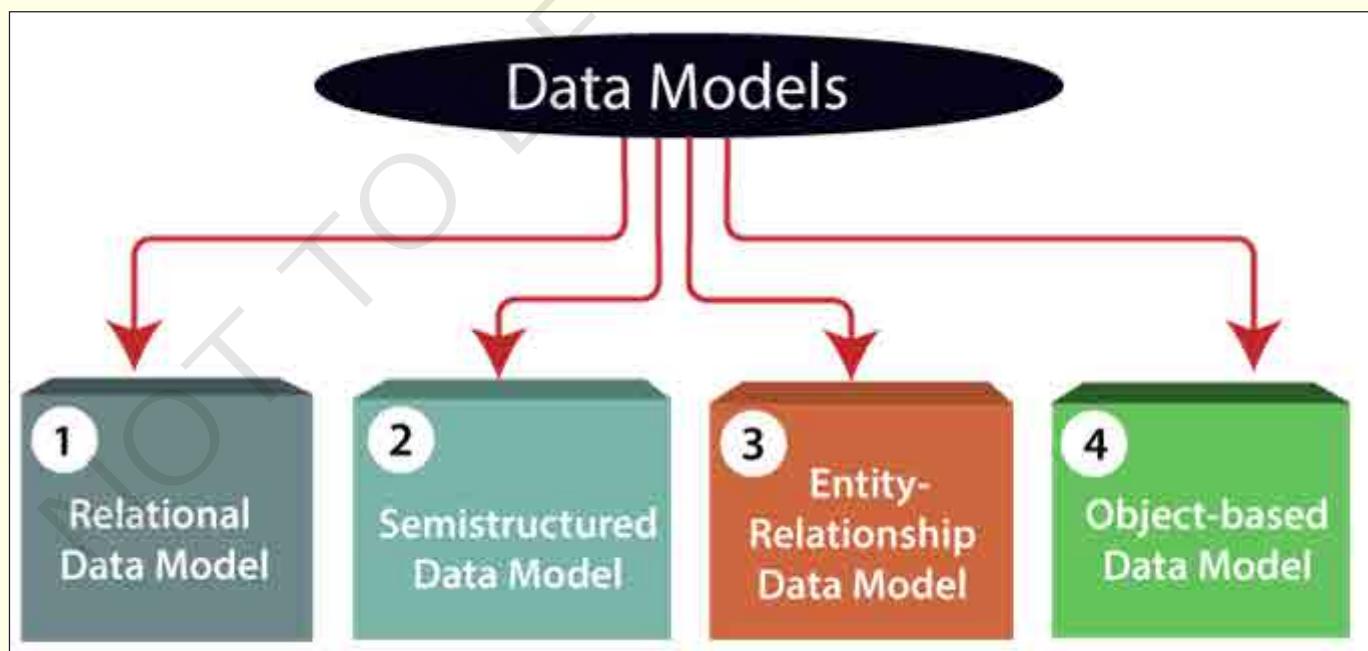
In general, databases store sets of data that can be queried for use in other applications. A database management system supports the development, administration and use of database platforms.

An RDBMS is a type of database management system (DBMS) that stores data in a row-based table structure which connects related data elements. An RDBMS includes functions that maintain the security, accuracy, integrity and consistency of the data. This is different than the file storage used in a DBMS.

Data Models

Data Model is the modeling of the data description, data semantics, and consistency constraints of the data. It provides the conceptual tools for describing the design of a database at each level of data abstraction. Therefore, there are following four data models used for understanding the structure of the database:

- 1 **Relational Data Model:** This type of model designs the data in the form of rows and columns within a table. Thus, a relational model uses tables for representing data and in-between relationships. Tables are also called relations. This model was initially described by Edgar F. Codd, in 1969. The relational data model is the widely used model which is primarily used by commercial data processing applications.



- 2 Entity-Relationship Data Model:** An ER model is the logical representation of data as objects and relationships among them. These objects are known as entities, and relationship is an association among these entities. This model was designed by Peter Chen and published in 1976 papers. It was widely used in database designing. A set of attributes describe the entities. For example, student_name, student_id describes the 'student' entity. A set of the same type of entities is known as an 'Entity set', and the set of the same type of relationships is known as 'relationship set'.
- 3 Object-based Data Model:** An extension of the ER model with notions of functions, encapsulation, and object identity, as well. This model supports a rich type system that includes structured and collection types. Thus, in 1980s, various database systems following the object-oriented approach were developed. Here, the objects are nothing but the data carrying its properties.
- 4 Semistructured Data Model:** This type of data model is different from the other three data models (explained above). The semistructured data model allows the data specifications at places where the individual data items of the same type may have different attributes sets. The Extensible Markup Language, also known as XML, is widely used for representing the semistructured data. Although XML was initially designed for including the markup information to the text document, it gains importance because of its application in the exchange of data.

Concept of DBA

There are many different types of databases which slightly differ from one another. While DBA are general database administrators that know how handle most things, there are DBAs who are experienced in managing a specific database. MySQL DBA is a database administrator who is an expert in managing MySQL databases.

MySQL is one of the most well-known, widely used, and efficient technologies in today's big data ecosystem. Basic knowledge of MySQL is essential for anyone involved with enterprise data or general IT, and even those who are unfamiliar with relational systems can quickly kickstart data storage systems using MySQL. The open source and relational nature of this database are its biggest advantages, being the reason for its use in both larger and smaller businesses.

That being said, and although MySQL database is considered well documented and has a large community to support it, there are still a few things that need to be taken into consideration when using MySQL database.

Database Users: Sometimes you want to manage a database in MySQL. In that case, we need to see the list of all user's accounts in a database. Most times, we assume that there is a SHOW USERS command similar to SHOW DATABASES, SHOW TABLES, etc. for displaying the list of all users available in the database server. Unfortunately, MySQL database does not have a SHOW USERS command to display the list of all users in the MySQL server. We can use the following query to see the list of all user in the database server:

```
mysql> Select user from mysql.user;
```

After the successful execution of the above statement, we will get the user data from the user table of the MySQL database server.

Database Schema

The mysql schema is the system schema. It contains tables that store information required by the MySQL server as it runs. A broad categorization is that the mysql schema contains data dictionary tables that store database object metadata, and system tables used for other operational purposes. The following discussion further subdivides the set of system tables into smaller categories.

- Data Dictionary Tables
- Grant System Tables
- Object Information System Tables
- Log System Tables
- Server-Side Help System Tables
- Time Zone System Tables
- Replication System Tables
- Optimizer System Tables
- Miscellaneous System Tables

The remainder of this section enumerates the tables in each category, with cross references for additional information. Data dictionary tables and system tables use the InnoDB storage engine unless otherwise indicated.

mysql system tables and data dictionary tables reside in a single InnoDB tablespace file named mysql.ibd in the MySQL data directory. Previously, these tables were created in individual tablespace files in the mysql database directory.

Data-at-rest encryption can be enabled for the mysql system schema tablespace. For more information, see Section 15.13, "InnoDB Data-at-Rest Encryption".

Designing database using normalization rules, various datatypes, data integrity, DDL, DML & DCL Statements Enforcing Primary key and Foreign key

Objectives: At the end of this lesson you shall be able to

- database Normalization in MySQL
- various datatypes
- data integrity
- DDL,DML&DCL Statements Database Normalization in MySQL
- enforcing Primary key and Foreign key
- adding Indices.

Database Normalization in MySQL

Database Normalization is the most important factor in Database design or Data modeling. Database Normalization is the process to eliminate data redundancies and store the data logically to make data management easier. Database relationships and keys are useful in the Database Normalization process. The Database Normalization was developed by E.F.Codd. In the database normalization process, there are series of rules called Normal Forms.

There are mainly 6 types of Normal Forms: First Normal Form (1NF), second Normal Form (2NF), Third Normal Form (3NF), Fourth Normal Form (4NF), Fifth Normal Form (5NF), and Boyce Codd Normal Form (BCNF). But majorly we used up to third normal form in our database design.

- 1 First Normal Form (1NF): In the first normal form, each column must contain only one value and no table should store repeating groups of related data.
- 2 Second Normal Form (2NF): In the second normal form, first the database must be in the first normal form, it should not store duplicate rows in the same table. And if there are duplicate values in the row, they should be stored in their own separate tables and linked to the table using foreign keys. The ideal way to a database in second normal form is to create one to many relationship tables.
- 3 Third Normal Form (3NF): In the third normal form, the database is already in the third form, if it is in the second normal form and every non-key column is mutually independent. Identify any columns in the table that are interdependent and break those columns into their own separate tables.
- 4 Boyce Codd Normal Form (BCNF): It is the highest form of the third normal form which deals with different types of anomalies that are not handled by the 3NF.

Various Data types

Data Integrity

Data integrity refers to the consistency, accuracy, and reliability of the data stored in any database or a warehouse. The data with a complete structure having all characteristics accurate is said to be data with integrity.

There are many aspects to the integrity of data like physical integrity, which involves storing and collecting the data authentically, then comes the logical integrity, which includes checking whether the information is relevant and accurate in the specific context or not so overall all the rules which are required to maintain the quality of data comes under data integrity. Without integrity and accuracy, all your collected data is useless to the company, so it's essential to ensure data protection and its accuracy to increase its performance and stability. Corrupted data can also damage your business sometimes.

It's effortless to alter data because data is not static. The information you receive comes from various places, and many things can change your data from the site it has been created. In addition, it can be transferred to other devices, altered, and updated by whomever and whenever required.

And ensuring data integrity is not just a one-step process. It has to be checked and confirmed at every step, starting from the model's design to its final output.

DDL , DML and DCL Statements: Structured Query Language(SQL) as we all know is the database language by the use of which we can perform certain operations on the existing database and also we can use this language to create a database. SQL uses certain commands like Create, Drop, Insert, etc. to carry out the required tasks.

These SQL commands are mainly categorized into four categories as:

- 1 DDL - Data Definition Language
- 2 DML - Data Manipulation Language
- 3 DCL - Data Control Language

DDL (Data Definition Language)

DDL or Data Definition Language actually consists of the SQL commands that can be used to define the database schema. It simply deals with descriptions of the database schema and is used to create and modify the structure of database objects in the database. DDL is a set of SQL commands used to create, modify, and delete database structures but not data. These commands are normally not used by a general user, who should be accessing the database via an application.

List of DDL commands

- **CREATE:** This command is used to create the database or its objects (like table, index, function, views, store procedure, and triggers).
- **DROP:** This command is used to delete objects from the database.
- **ALTER:** This is used to alter the structure of the database.
- **TRUNCATE:** This is used to remove all records from a table, including all spaces allocated for the records are removed.
- **COMMENT:** This is used to add comments to the data dictionary.
- **RENAME:** This is used to rename an object existing in the database.

DML(Data Manipulation Language)

The SQL commands that deals with the manipulation of data present in the database belong to DML or Data Manipulation Language and this includes most of the SQL statements. It is the component of the SQL statement that controls access to data and to the database. Basically, DCL statements are grouped with DML statements.

List of DML commands

- **INSERT :** It is used to insert data into a table.
- **UPDATE:** It is used to update existing data within a table.
- **DELETE :** It is used to delete records from a database table.
- **LOCK:** Table control concurrency.
- **CALL:** Call a PL/SQL or JAVA subprogram.
- **EXPLAIN PLAN:** It describes the access path to data.

DCL (Data Control Language)

DCL includes commands such as GRANT and REVOKE which mainly deal with the rights, permissions, and other controls of the database system.

List of DCL commands

- **GRANT:** This command gives users access privileges to the database.
- **REVOKE:** This command withdraws the user's access privileges given by using the GRANT command.

Primary Key

Primary Key Constraint is defined by creating a primary key on a table. The value in primary key must uniquely identify each row in the table. Primary Key Constraint enforces row integrity.

Consider the following facts when defining a primary key:

- When you create a primary key on a table, a unique index (and hence unique constraint) is automatically and implicitly created which enforces uniqueness of data in this column.
- Primary key column cannot contain NULL values. Database engine implicitly creates a NOT NULL constraint on the primary key column.
- A table can only have one primary key defined on it.
- A primary key can be formed by a single column or a composition of multiple columns (i.e. composite primary key).
- If you insert a duplicated record in primary key, you'll get an error.

There are two types of primary key: artificial primary key and natural primary key.

Foreign Key

MySQL supports foreign keys, which permit cross-referencing related data across tables, and foreign key constraints, which help keep the related data consistent.

A foreign key relationship involves a parent table that holds the initial column values, and a child table with column values that reference the parent column values. A foreign key constraint is defined on the child table.

Insert and delete queries Update queries

Objectives: At the end of this lesson you shall be able to

- simple select queries.
-

Adding Indices

Queries concept of transaction

A transaction is a sequential group of database manipulation operations, which is performed as if it were one single work unit. In other words, a transaction will never be complete unless each individual operation within the group is successful. If any operation within the transaction fails, the entire transaction will fail.

Practically, you will club many SQL queries into a group and you will execute all of them together as a part of a transaction.

Properties of Transactions

Transactions have the following four standard properties, usually referred to by the acronym ACID ?

- **Atomicity** - This ensures that all operations within the work unit are completed successfully; otherwise, the transaction is aborted at the point of failure and previous operations are rolled back to their former state.
- **Consistency** - This ensures that the database properly changes states upon a successfully committed transaction.
- **Isolation** - This enables transactions to operate independently on and transparent to each other.
- **Durability** - This ensures that the result or effect of a committed transaction persists in case of a system failure.

Using the Number, Date and Character functions, group by having, sub query

Objectives: At the end of this lesson you shall be able to

- state the sub query.

MySQL Joining Tables

A JOIN clause is used to combine rows from two or more tables, based on a related column between them.

Let's look at a selection from the "Orders" table:

OrderID	CustomerID	OrderDate
10308	2	1996-09-18
10309	37	1996-09-19
10310	77	1996-09-20

Then, look at a selection from the "Customers" table:

CustomerID	CustomerName	ContactName	Country
1	Alfreds Futterkiste	Maria Anders	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mexico

Notice that the "CustomerID" column in the "Orders" table refers to the "CustomerID" in the "Customers" table. The relationship between the two tables above is the "CustomerID" column.

Then, we can create the following SQL statement (that contains an INNER JOIN), that selects records that have matching values in both tables:

Example

```
SELECT Orders.OrderID, Customers.CustomerName,
       Orders.OrderDate
```

```
FROM Orders
```

```
INNER JOIN Customers
       ON Orders.CustomerID=Customers.CustomerID;
```

Try it Yourself "and it will produce something like this:

OrderID	CustomerName	OrderDate
10308	Ana Trujillo Emparedados y helados	9/18/1996
10365	Antonio Moreno Taquería	11/27/1996
10383	Around the Horn	12/16/1996
10355	Around the Horn	11/15/1996
10278	Berglunds snabbköp	8/12/1996

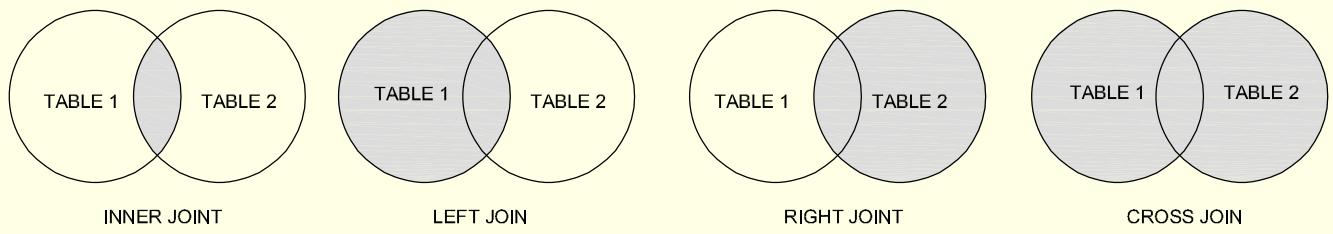
Supported Types of Joins in MySQL

- INNER JOIN: Returns records that have matching values in both tables
- LEFT JOIN: Returns all records from the left table, and the matched records from the right table
- RIGHT JOIN: Returns all records from the right table, and the matched records from the left table
- CROSS JOIN: Returns all records from both tables

SUB QUERIES

A subquery in MySQL is a query, which is nested into another SQL query and embedded with SELECT, INSERT, UPDATE or DELETE statement along with the various operators. We can also nest the subquery with another subquery. A subquery is known as the inner query, and the query that contains subquery is known as the outer query. The inner query executed first gives the result to the outer query, and then the main/outer query will be performed. MySQL allows us to use subquery anywhere, but it must be closed within parenthesis. All subquery forms and operations supported by the SQL standard will be supported in MySQL also.

Fig 1



CP20N1299911

The following are the rules to use subqueries

- Subqueries should always use in parentheses.
- If the main query does not have multiple columns for subquery, then a subquery can have only one column in the SELECT command.
- We can use various comparison operators with the subquery, such as >, <, =, IN, ANY, SOME, and ALL. A multiple-row operator is very useful when the subquery returns more than one row.
- We cannot use the ORDER BY clause in a subquery, although it can be used inside the main query.
- If we use a subquery in a set function, it cannot be immediately enclosed in a set function.

The following are the advantages of using subqueries

- The subqueries make the queries in a structured form that allows us to isolate each part of a statement.
- The subqueries provide alternative ways to query the data from the table; otherwise, we need to use complex joins and unions.
- The subqueries are more readable than complex join or union statements.

MySQL Subquery Syntax

The following is the basic syntax to use the subquery in MySQL:

SELECT column_list (s) **FROM** table_name

WHERE column_name OPERATOR

(SELECT column_list (s) **FROM** table_name [WHERE])

Functions used in Query

- 1 **sum:**
Calculates the sum of a set of values
Syntax: SUM(expression)
- 2 **average (avg):**
Returns the average value of an expression
Syntax :AVG(expression)
- 3 **MAX:**
Returns the maximum value in a set of values
Syntax: MAX(expression)
- 4 **MIN :**
Returns the minimum value in a set of values
Syntax: MIN(expression)
- 5 **Count :**
Returns the number of records returned by a select query
Syntax: COUNT(expression)

Connect a computer to a network and share Devices i.e. Printers, files, folders and drives

Objectives: At the end of this lesson you shall be able to

- local Networks.
 - communicating on a local network:
 - principles of Communications.
-

Communicating in a connected world

Local Networks.

A local area network (LAN) is a group of computers and peripheral devices that share a common communications line or wireless link to a server within a distinct geographic area. A local area network may serve as few as two or three users in a home office or thousands of users in a corporation's central office.

Communicating on a local network:

A peer-to-peer LAN directly connects two devices -- generally, workstations or personal computers -- together using an Ethernet cable. A client-server LAN consists of multiple endpoints and servers that are connected to a LAN switch. The switch directs communication streams between the multiple connected devices.

Principles of Communications:

Communication Professionals will sometimes refer to the 7 Principles of Communication; sadly, these are not standardised (so probably level 200!), but there are some common themes which were used to inform the Communication Competency within the Microsoft 365 Maturity Model. If you are not a Communications professional you may find this interpretation of the principles helpful as part of a communications strategy, maturity assessment or technology selection.

- 1 Effective
- 2 Comprehensive
- 3 Clarity
- 4 Attention and Style
- 5 Coherency
- 6 Timeliness and Urgency
- 7 Importance of Feedback

Ethernet networks

Objectives: At the end of this lesson you shall be able to

- define Ethernet
 - how to work Ethernet
 - define network built.
-

Ethernet is one of the original networking technologies, having been invented nearly 50 years ago. And yet, because of the simplicity by which the communications protocol can be deployed and its ability to incorporate modern advancements without losing backwards compatibility, Ethernet continues to reign as the de facto standard for computer networking.

At its core, Ethernet is a protocol that allows computers (from servers to laptops) to talk to each other over wired networks that use devices like routers, switches and hubs to direct traffic. Ethernet works seamlessly with wireless protocols, too.

Its ability to work within almost any environment has led to its universal adoption around the world. This is especially true because it allows organizations to use the same Ethernet protocol in their local area network (LAN) and their wide-area network (WAN). That means that it works well in data centers, in private or internal company networks, for internet applications and almost anything in between. It can even support the most complex forms of

networking, like virtual private networks (VPNs) and software-defined networking deployments.

Ethernet has no problem handling bandwidth-intensive applications such as video streaming or voice over IP applications. And on the other end, its simplicity also enables it to work with very tiny, relatively unsophisticated devices such as those that make up the Internet of Things (IoT), without any special configuration required.

How does Ethernet work?

Ethernet works by breaking up information being sent to or from devices, like a personal computer, into short pieces of different sized bits of information called frames. Those frames contain standardized information such as the source and destination address that helps the frame route its way through a network.

And because computers on a LAN typically share a single connection, Ethernet was built around the principle of CSMA/CD, or carrier-sense multiple access with collision detection. Basically, the protocol makes sure that the line

is not in use before sending any frames out. Today, that is far less important than it was in the early days of networking because devices generally have their own private connection to a network through a switch or node. And because Ethernet now operates using full duplex, the sending and receiving channels are also completely separate, so collisions can't actually occur over that leg of their journey.

Other than when encountering a collision situation, there is no error correction in Ethernet, so communications need to rely on advanced protocols to ensure that everything is being transmitted perfectly. However, Ethernet still provides

the basis for most internet and digital communications, and also integrates easily with most higher-level protocols, so that is almost never an issue these days.

How are network built?

Computer networks are formed by using nodes and connections (links) that enable teams to communicate with each other. Each node uses a network interface card that defines how it interacts with the network. Each network card has its own numeric address.

End-User Devices and local networks

Objectives: At the end of this lesson you shall be able to

- define communicate network
- types of Node Devices in a Computer Network
- functions of End Devices.

The network devices that people are most familiar with are called end devices. All computers connected to a network that participate directly in network communication are classified as hosts. These devices form the interface between users and the underlying communication network.

End user node

There are many examples of end user devices, including desktop computers, laptops, printers, tablets, and servers. The most common terms for these end user devices are host, and end user node; we will use end user node throughout this chapter.

Types of Node Devices in a Computer Network:

End devices and Intermediary Devices

We know a computer network is a term used to refer to any group (or system) of interconnected nodes (computers, printers, or any other devices) connected by communication links known as transmission media (or channels) usually meant for the exchange of information and resource sharing. In a computer network, a node is any device that is capable of sending or receiving data, to and from other nodes at definite and desired flow rates securely and reliably.

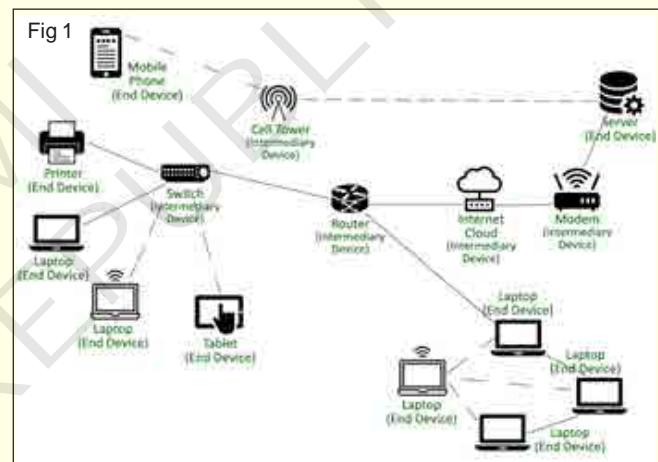
On the basis of functionality and usage, the node devices can be broadly classified into the following types -

- 1 End Devices
- 2 Intermediary Devices

End Devices

End devices are the node devices that serve as a source point or a destination point in the communication that occurs on a computer network. With the coming advancements in computer networking systems, we have nodes that can act as a client, a server, or both. The rest of the network is built around these end devices to establish communication links between them. Software installed on the node devices determines the role they offer to play in a computer network. According to their usage, the end

devices can be broadly grouped into the following categories - (Fig 1)



- **End Devices as Clients** - Perform the tasks of requesting data, displaying received data, etc. meant usually for the usage by end clients.
- **End Devices as Servers** - Equipped with programs that provide information and services such as webpages or e-mails to other nodes (or hosts) on the network.

Functions of End Devices

- 1 They serve as the originator of the data or information that flows through the network.
- 2 Act as an interface between end-users (humans) and the communication network having several node devices.

Examples of the End Devices

Work Stations, Laptops, Desktop Computers, Printers and Scanners, Servers (File Servers, Web Servers), Mobile Phones, Tablets, Smart Phones, etc.

Intermediary Devices

Intermediary devices are node devices that are designed to forward the data from one side to another side in a computer network. These intermediary devices work as a connecting medium (along with other services being offered) for other nodes and handle the tasks in the background ensuring that the data flows effectively at desired flow rates across the entire computer network.

The intermediary devices for the management of the data flowing through them use various addressing systems such as IP Address, MAC Address, and Port Numbers (or Port Address) along with the information about the network interconnections. Further various types of switching in the computer networks determine the path that messages take through the network during the communication.

Functions of Intermediary Devices

- Signal damping is a common phenomenon that is overcome via regeneration and retransmission of the data signals done by these devices.
- To ensure successful transfers these devices keep record of information of source address, destination address or different pathways existing through the network depending upon the switching technique being employed.
- They also detect faults and errors effectively using redundancy bits, etc. and notify the devices to further ensure fault tolerance by performing corrections while transferring data in a computer network.
- Maintenance of a definite desired flow control and response timeout during the entire course of the communication.
- Setup, grouping and direction of messages (or packets) according to their priorities is done by these devices.
- Provide network security by permitting or denying the flow of data, based on security settings.

Examples of the intermediary devices

Hubs, switches, wireless access points, and other devices used for accessing the network, file servers, web servers, print servers, modems, devices using for internetworking such as routers, bridges, repeaters, and security firewalls, etc.

Local network

Networks can be categorized depending on size, complexity, level of security, or geographical range. We will discuss some of the most popular topologies based on geographical spread.

PAN (Fig 2)

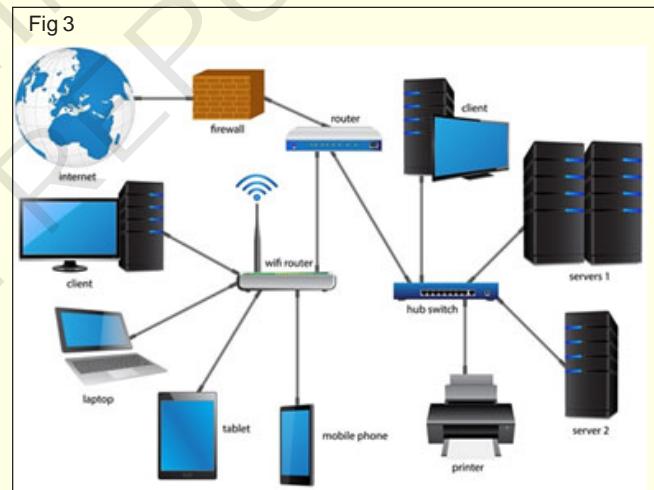
PAN is the acronym for Personal Area Network. PAN is the interconnection between devices within the range of a person's private space, typically within a range of 10 metres. If you have transferred images or songs from your laptop to mobile or from mobile to your friend's mobile using Bluetooth, you have set up and used a personal area network.



A person can connect her laptop, smart phone, personal digital assistant and portable printer in a network at home. This network could be fully Wi-Fi or a combination of wired and wireless.

LAN (Fig 3)

LAN or Local Area Network is a wired network spread over a single site like an office, building or manufacturing unit. LAN is set up to when team members need to share software and hardware resources with each other but not with the outside world. Typical software resources include official documents, user manuals, employee handbook, etc. Hardware resources that can be easily shared over the network include printer, fax machines, modems, memory space, etc. This decreases infrastructure costs for the organization drastically.



A LAN may be set up using wired or wireless connections. A LAN that is completely wireless is called Wireless LAN or WLAN.

MAN (Fig 4)

MAN is the acronym for Metropolitan Area Network. It is a network spread over a city, college campus or a small region. MAN is larger than a LAN and typically spread over several kilometres. Objective of MAN is to share hardware and software resources, thereby decreasing infrastructure costs. MAN can be built by connecting several LANs.

The most common example of MAN is cable TV network.

Fig 4



WAN (Fig 5)

WAN or Wide Area Network is spread over a country or many countries. WAN is typically a network of many LANs, MANs and WANs. Network is set up using wired or wireless connections, depending on availability and reliability.

The most common example of WAN is the Internet.

Fig 5



Network communication

- Basics of network communication
- Define Home network
- Define Wi-fi
- Introduction to LAN

Basics of network communication

This section looks at network communication from a very high level and defines some terms used throughout the information. For more detailed information on z/OS® network communication and TCP/IP sockets, see z/OS Communications Server: IP Configuration Guide and z/OS Communications Server: IP Programmer's Guide and Reference. For more detailed information on IPv6 network communication and AF_INET6 sockets, see z/OS Communications Server: IPv6 Network and Appl Design Guide.

Network communication, or internetworking, defines a set of protocols (that is, rules and standards) that allow application programs to talk with each other without regard to the hardware and operating systems where they are run. Internetworking allows application programs to communicate independently of their physical network connections.

The internetworking technology called TCP/IP is named after its two main protocols: Transmission Control Protocol (TCP) and Internet Protocol (IP). To understand TCP/IP, you should be familiar with the following terms:

Client

A process that requests services on the network.

Server

A process that responds to a request for service from a client.

Datagram

The basic unit of information, consisting of one or more data packets, which are passed across an Internet at the transport level.

Packet

The unit or block of a data transaction between a computer and its network. A packet usually contains a network header, at least one high-level protocol header, and data blocks. Generally, the format of data blocks does not affect how packets are handled. Packets are the exchange medium used at the Internetwork layer to send data through the network.

How do devices on the Internet communicate with each other?

Ethernet is a standard protocol that allows any number of computers to communicate with one another. These machines need to be connected to one another through Ethernet cable, or "Category 5" wiring.

What is Home Networking?

A home network is a group of devices - such as computers, game systems, printers, and mobile devices - that connect to the Internet and each other. Home networks connect in two ways:

- A wired network, which connects devices like printers and scanners with cables
- A wireless network, which connects devices like tablets and e-readers without cables

Why Set Up a Home Network?

There are many reasons to establish a home network. Here are just a few of the things home networking allows you to do:

- Connect to the Internet from multiple computers, game systems, mobile devices, and more.
- Access files and folders on all devices connected to the network.
- Print from multiple computers on a single printer.
- Manage security settings for all networked devices in one place.

If you're ready to try home networking, read below to find out what you'll need to get started and to see the options available for your network.

What You Need to Set Up a Home Network

To set up home networking, you'll need the following:

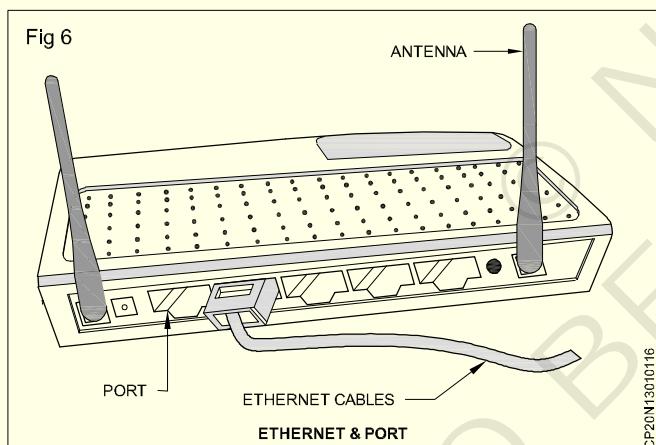
- Xfinity Internet Service subscription (or subscription to another Internet provider)
- A modem, which connects to the Internet, and a router, which connects your devices to each other and to the Internet through your modem (or a gateway, which functions as both a modem and a router)
- A computer or other device to connect to the network

See Activate a Wireless Gateway for more information.

The Wireless Gateway 1 (model numbers TG852G, TG862G, SMCD3GNV, TC8305C) and Wireless Gateway 2 (model number DPC3939) function as an all-in-one modem, router, and phone device. They automatically provide users with the best security settings available for a home network. See What is a Wireless Gateway? for more information.

Wireless Home Network: A wireless network, often called Wi-Fi, connects devices to each other and to the Internet without using cables. See What Is WiFi and How Do I Get It? for more information.

Wired Home Network (Fig 6): A wired home network connects devices to each other and to the Internet using Ethernet cables.



There are several benefits to having a wired home network:

- Faster and more reliable connection to the Internet
- Increased security, as no outside users can access your network
- Easier set-up and troubleshooting than wireless connections

Mixed Home Network: Many people find that a mix of wireless and wired networking meets their needs best. For instance, devices that stream movies benefit from the quicker and more stable wired connection. Devices like laptops or tablets, however, benefit from the mobility available with a wireless connection.

Both the Wireless Gateway 1 and Wireless Gateway 2 come with wireless capability and four Ethernet ports, allowing you to connect devices with and without cables at the same time.

Home Networking Glossary

Below is a list of common home networking terms. Take a moment to familiarize yourself with these definitions, as they'll likely be mentioned in other help articles.

Wireless network - A group of devices (computers, game systems, e-readers, etc.) connected to each other and to the Internet without cables, also called Wi-Fi.

Modem - A device that connects routers or computers to the Internet through a cable.

Router - A device that transmits a wireless signal and connects devices to each other and the Internet through the modem.

Gateway - A device that functions as both a modem and a router, like the Wireless Gateway 1 or Wireless Gateway 2.

Signal range - The area in which a wireless-capable device can pick up the wireless signal and connect to the wireless network.

Administration site - The online site used to view and change the settings of a router and wireless network, called the Admin Tool for the Wireless Gateway 1 and Wireless Gateway 2.

Router username and password - The login information used to access the administration site.

Network Name (SSID) - The name of a wireless network.

Network Password (Key) - The password used to connect to a wireless network.

Wi-Fi Definition

Wi-Fi is a wireless technology used to connect computers, tablets, smartphones and other devices to the internet.

Wi-Fi is the radio signal sent from a wireless router to a nearby device, which translates the signal into data you can see and use. The device transmits a radio signal back to the router, which connects to the internet by wire or cable.

What is a Wi-Fi network?

A Wi-Fi network is simply an internet connection that's shared with multiple devices in a home or business via a wireless router. The router is connected directly to your internet modem and acts as a hub to broadcast the internet signal to all your Wi-Fi enabled devices. This gives you flexibility to stay connected to the internet as long as you're within your network coverage area.

What does Wi-Fi stand for?

The term was created by a marketing firm because the wireless industry was looking for a user-friendly name to refer to some not so user-friendly technology known as IEEE 802.11 and the name stuck. Wi-Fi, often referred to as WiFi, wifi, wi-fi or wi fi, is often thought to be short for Wireless Fidelity and the organization that paid for the marketing firm is sometimes referred to as the Wireless Fidelity Alliance Inc.

How does Wi-Fi work?

Wi-Fi uses radio waves to transmit data from your wireless router to your Wi-Fi enabled devices like your TV, smartphone, tablet and computer. Because they communicate with each other over airwaves, your devices and personal information can become vulnerable to hackers, cyber-attacks and other threats. This is especially true when you connect to a public Wi-Fi network at places like a coffee shop or airport. When possible, it's best to connect to a wireless network that is password-protected or a personal hotspot.

Types of Wi-Fi connections

Your options for connecting wirelessly at home are growing as mobile networks expand into the home internet realm. As with internet service, there are advantages and disadvantages to each type of wireless connection, such as speed and strength of signal. We've outlined a few of them here.

Wireline/router

Most homes use a wireless router to access the internet these days. The pros include convenience of setup, mobility within range of the Wi-Fi access point (router) and the ability to connect multiple devices. The cons: limited bandwidth and reduced speed as more devices are connected to the same Wi-Fi network, as well as potential interference from other electromagnetic devices in the home.

Mobile hotspot or jetpack

Mobile and dedicated hotspots are becoming a more popular way of connecting safely on the go. Two common hotspot devices are your smartphone and a jetpack. Most any smartphone or tablet today can be used as a temporary hotspot and is a great option if you occasionally need it. It's easy to use and doesn't require buying extra devices, but it can zap your battery life and data pretty fast. On the other hand, a jetpack acts as a dedicated mobile hotspot that picks up a signal from cell towers in your area just like your smartphone. More devices can connect to it and it offers a greater range of Wi-Fi. And because it's a separate device, your smartphone battery power doesn't get touched. The downfall is having to buy the jetpack and a separate plan.

LTE Home Internet

If you live in a rural area where your internet options are limited, 4G LTE Home Internet is worth considering. It offers high speed internet service delivered over cell phone towers and mobile networks with average download speeds around 25 Mbps. The advantages of LTE over satellite are better speeds and reliability depending on your carrier.

5G Home Internet: 5G Home Internet (Fixed Wireless Access) is now in more places around the country.* Even with multiple devices connected, it's reliable and fast enough to power your whole home. Fixed wireless access has a simple plug and play setup, which means there are no messy wires and no need to wait for a technician to come to your home for setup.

How to get Wi-Fi at home

As mentioned, there are several ways to get wireless service at your home and most depend on geographical location and availability. (See what Wi-Fi home internet is available in your area.) The majority of urban and suburban areas offer most of these services, with 5G Home Internet right around the corner. Rural areas will most likely offer satellite and 4G LTE Home Internet. If you have a wired internet service, you'll be able to set up your own Wi-Fi network at home. By connecting a router to your modem, you can share your internet connection with all your Wi-Fi enabled devices within range. If your home has two levels, concrete walls or random dead zones, adding a Wi-Fi extender that relays the wireless signal to these areas can make a big difference.

Keep in mind that as the number of your mobile devices grows, so does the demand for bandwidth. To keep your devices running at top speeds, you may need to upgrade your internet speed plan. Verizon offers several internet services ranging from DSL and fiber to 5G Home Internet, depending on your location. For better reliability and Verizon's fastest download speeds, check if Fios Internet or 5G Home Internet are available at your address.

See what Wi-Fi internet is available in your area.

Introduction to LAN

A local area network (LAN) consists of a series of computers linked together to form a network in a circumscribed location. The computers in a LAN connect to each other via TCP/IP ethernet or Wi-Fi. A LAN is normally exclusive to an organization, such as a school, office, association or church.

What is LAN device?

A local area network (LAN) is a group of computers and peripheral devices that share a common communications line or wireless link to a server within a distinct geographic area. A local area network may serve as few as two or three users in a home office or thousands of users in a corporation's central office.

Types of network devices

- Hub.
- Switch.
- Router.
- Bridge.
- Gateway.
- Modem.
- Repeater.
- Access Point.

Types of Local Area Network (LAN)

- Peer to peer (P2P) LAN. ...
- Token ring LAN. ...
- Token bus LAN. ...

- Wired LAN. ...
- Cloud-managed LAN. ...
- Public internet. ...

- Wired end-user devices. ...
- Mobile end-user devices.

LAN vs. MAN vs. WAN (Fig 7)

Fig 7	LAN	MAN	WAN
GEOGRAPHICAL AREA	Small area, 1 to 5 km	City limits, 50 to 60 km	Global, up to 1000s of km
TYPICAL CUSTOMERS	Schools and colleges, offices, small industries and SMBs	Mid-market to large enterprises, city governments, business chains and financial institutions	Large nationwide or global enterprises
TECHNICAL ASPECTS	Ethernet and Ethernet switches; some L3 switching and routing	Ethernet switching on the LAN; Metro Ethernet on the MAN	MPLS, VPLS, SONET and satellite technologies

Internet concepts

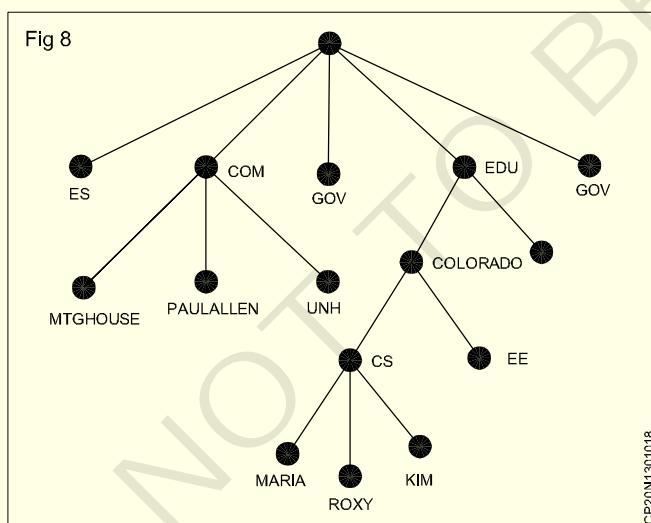
- Define www
- WWW Architecture
- User Interface and Applications

WWW stands for World Wide Web. A technical definition of the World Wide Web is : all the resources and users on the Internet that are using the Hypertext Transfer Protocol (HTTP).

The World Wide Web is the universe of network-accessible information, an embodiment of human knowledge.

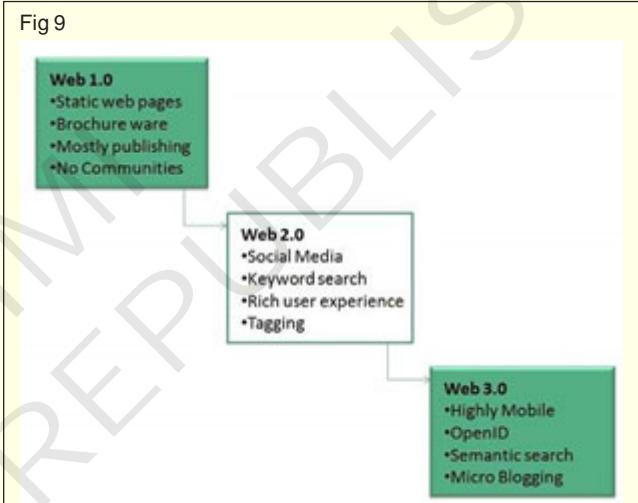
In simple terms, The World Wide Web is a way of exchanging information between computers on the Internet, tying them together into a vast collection of interactive multimedia resources.

Internet and Web is not the same thing: Web uses internet to pass over the information. (Fig 8)



Evolution: World Wide Web was created by Timothy Berners Lee in 1989 at CERN in Geneva. World Wide Web came into existence as a proposal by him, to allow researchers to work together effectively and efficiently at CERN. Eventually it became World Wide Web.

The following diagram briefly defines evolution of World Wide Web: (Fig 9)



WWW Architecture

WWW architecture is divided into several layers as shown in the following diagram:

Identifiers and Character Set

Uniform Resource Identifier (URI) is used to uniquely identify resources on the web and UNICODE makes it possible to build web pages that can be read and write in human languages.

Syntax

XML (Extensible Markup Language) helps to define common syntax in semantic web.

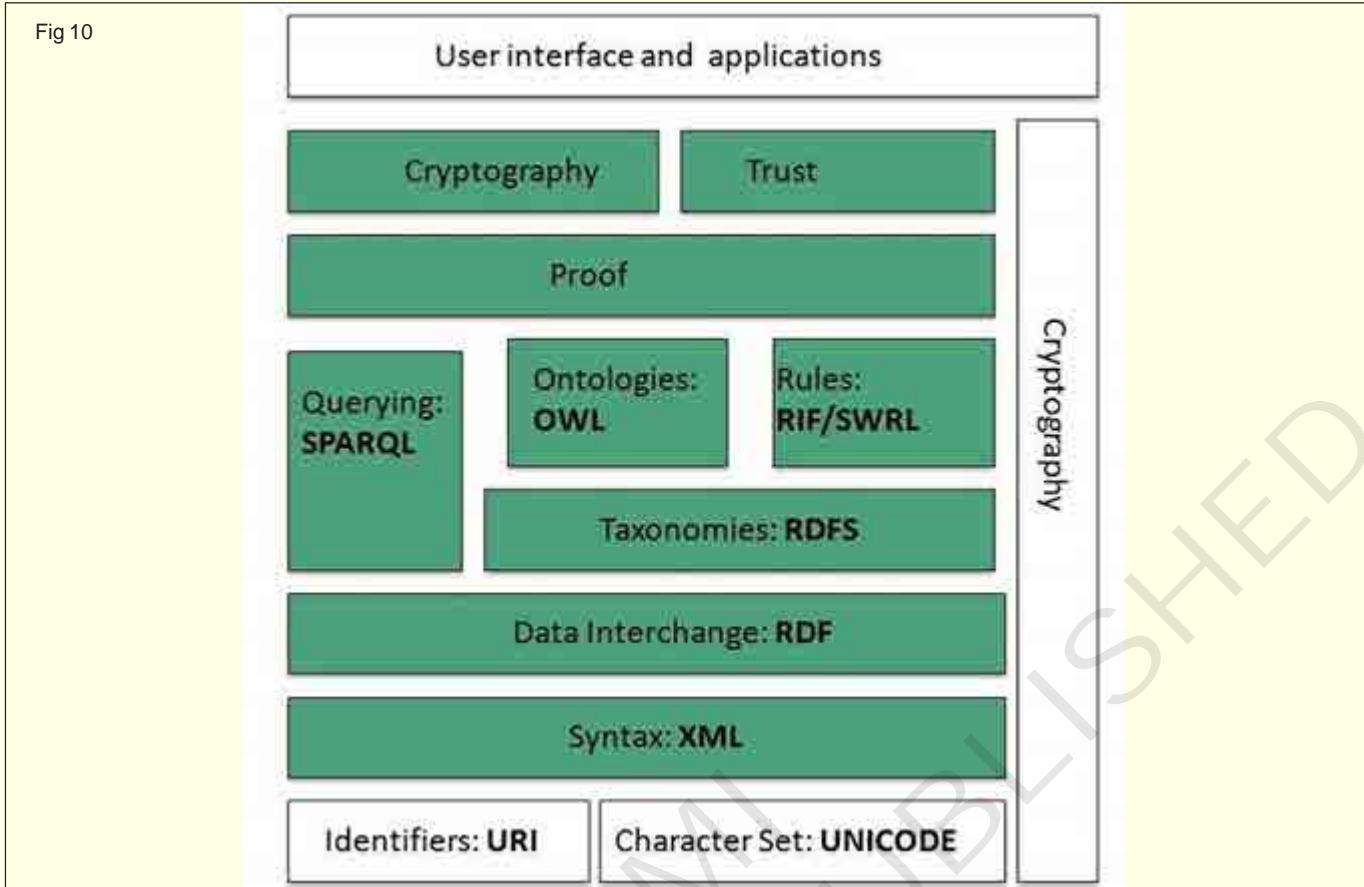
Data Interchange

Resource Description Framework (RDF) framework helps in defining core representation of data for web. RDF represents data about resource in graph form.

Taxonomies

RDF Schema (RDFS) allows more standardized description of taxonomies and other ontological constructs.

Fig 10



Ontologies (Fig 10)

Web Ontology Language (OWL) offers more constructs over RDFS. It comes in following three versions:

- OWL Lite for taxonomies and simple constraints.
- OWL DL for full description logic support.
- OWL for more syntactic freedom of RDF

Rules

RIF and SWRL offers rules beyond the constructs that are available from RDFS and OWL. Simple Protocol and RDF Query Language (SPARQL) is SQL like language used for querying RDF data and OWL Ontologies.

Proof

All semantic and rules that are executed at layers below Proof and their result will be used to prove deductions.

Cryptography

Cryptography means such as digital signature for verification of the origin of sources is used.

User Interface and Applications

On the top of layer User interface and Applications layer is built for user interaction.

WWW Operation

WWW works on client- server approach. Following steps explains how the web works:

- 1 User enters the URL (say, <http://www.tutorialspoint.com>) of the web page in the address bar of web browser.

- 2 Then browser requests the Domain Name Server for the IP address corresponding to www.tutorialspoint.com.
- 3 After receiving IP address, browser sends the request for web page to the web server using HTTP protocol which specifies the way the browser and web server communicates.
- 4 Then web server receives request using HTTP protocol and checks its search for the requested web page. If found it returns it back to the web browser and close the HTTP connection.
- 5 Now the web browser receives the web page, It interprets it and display the contents of web page in web browser's window.

Future

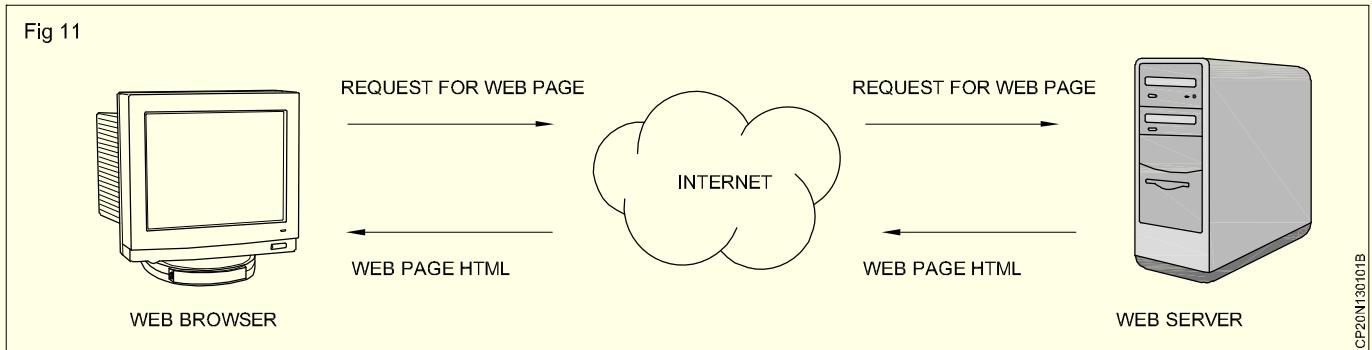
There had been a rapid development in field of web. It has its impact in almost every area such as education, research, technology, commerce, marketing etc. So the future of web is almost unpredictable.

Apart from huge development in field of WWW, there are also some technical issues that W3 consortium has to cope up with.

User Interface (Fig 11)

Work on higher quality presentation of 3-D information is under development. The W3 Consortium is also looking forward to enhance the web to full fill requirements of global communities which would include all regional languages and writing systems.

Fig 11



Technology

Work on privacy and security is under way. This would include hiding information, accounting, access control, integrity and risk management.

Architecture

There has been huge growth in field of web which may lead to overload the internet and degrade its performance. Hence more better protocol are required to be developed.

Whats is E-Mail(Electronic Mail)?

E-Mail (Electronic Mail) is the exchange of computer - stored messages from one user to one or more recipients via the internet. E-Mails are a fast, inexpensive and accessible way to communicate for business or personal use.

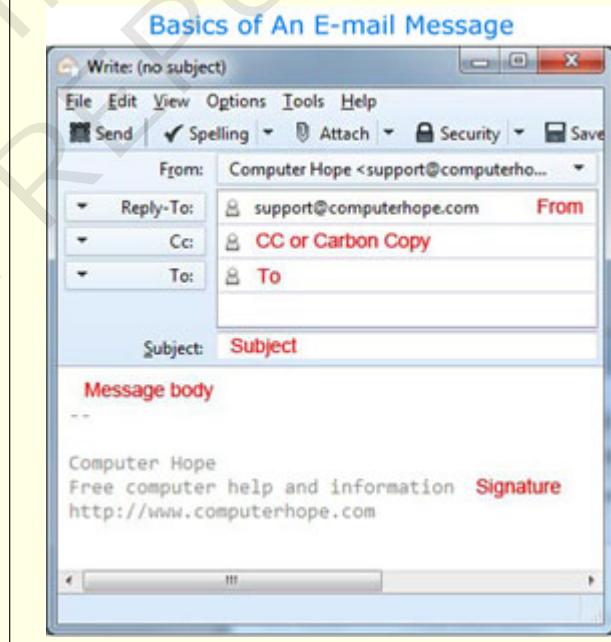
Writing an e-mail (Fig 12)

When writing an e-mail message, it should look something like the example window below. As you can see, several fields are required when sending an e-mail:

- The To field is where the e-mail address of the person receiving the e-mail is placed.
- The From field should contain your e-mail address.
- If you are replying to a message, the To: and From: fields are automatically filled out. If it's a new message, you need to specify the recipients in the To: field by selecting them from your contact list or by typing the e-mail addresses. If you enter more than one recipient (e.g., group e-mail), the addresses should be separated by a comma and a space or by pressing the Tab key.
- The Subject should consist of a few words describing the e-mail's contents. The subject lets the recipient see what the e-mail is about, without opening and reading the full e-mail. This field is optional.

- The CC ("Carbon Copy") field lets you specify recipients who are not direct addressees (listed in the "To" field). For instance, you can address an e-mail to Jeff and CC Linda and Steven. Although the e-mail is addressed to Jeff, Linda and Steven also receive a copy and everyone can see who received the e-mail. This field is optional.
- The BCC ("blind carbon copy") field is similar to CC, except the recipients are secret. Each BCC recipient receives the e-mail, but does not see who else received a copy. The addressees (anyone listed in the "To" field) remain visible to all recipients. This field is optional.
- Finally, the Message Body is the location you type your main message. It often contains your signature at the bottom; similar to a handwritten letter.

Fig 12



Web designing

Objectives : At the end of this lesson you shall be able to

- explain web sites and web pages
- explain static and dynamic web pages
- explain HTML, DHTML and XML
- explain the concept of web hosting, web server, application server and database server.

Introduction

The World Wide Web (WWW) was created in 1990 by CERN physicist Tim Berners-Lee. On 30 April 1993, CERN announced that the World Wide Web would be free to use for anyone. Before the introduction of HTML and HTTP, other protocols such as File Transfer Protocol and the Gopher Protocol were used to retrieve individual files from a server. These protocols offer a simple directory structure which the user navigates and chooses files to download. Documents were most often presented as plain text files without formatting, or were encoded in word processors formats.

Websites

A website, also written as Web site, web site, or simply site, is a set of related web page containing content (media) such as written language, Image, video, sound, etc. A website is hosted on at least one web server, accessible via a network such as the internet or a private local area network through an Internet address known as a uniform resource locator. All publicly accessible websites collectively constitute the world wide web.

A webpage is a document, typically written in plain text interspersed with formatting instructions of Hypertext Markup Language (HTML) XHTML. A webpage may incorporate elements from other websites with suitable HTML anchor.

Web pages are accessed and transported with the Hypertext Transfer Protocol (HTTP), which may optionally employ encryption secure, HTTPS to provide security and privacy for the user of the webpage content. The user's application, often a web browser, renders the page content according to its HTML Mark-up instructions onto a Computer monitor.

The pages of a website can usually be accessed from a simple Uniform Resource Locator (URL) called the web address. The URLs of the pages organize them into a hierarchy, although hyperlink between them conveys the reader's perceived sitemap and guides the reader's navigation of the site which generally includes a Home page with most of the links to the site's web content, and a supplementary about page, contact page and link page.

Some websites require a subscription to access some or all of their content. Examples of subscription websites include many business sites, parts of news websites, Academic journal websites, gaming websites, file-sharing websites, Internet forum, web-based Email, Social networking websites, websites providing real-time Stock

market data, and websites providing various other services (e.g., websites offering storing and/or sharing of images, files and so forth).

A website may be

- Personal website
- Commercial website
- E-Government
- Non-profit organization website.

Static Website

A static website is one that has web pages stored on the server in the format that is sent to a client web browser. It is primarily coded in Hypertext Markup Language (HTML).

Simple forms or marketing examples of websites, such as classic website, a five-page website or a brochure website are often static websites, because they present pre-defined, static information to the user. This may include information about a company and its products and services through text, photos, animations, audio/video and interactive menus and navigation.

This type of website usually displays the same information to all visitors. Similar to handing out a printed brochure to customers or clients, a static website will generally provide consistent, standard information for an extended period of time. Although the website owner may make updates periodically, it is a manual process to edit the text, photos and other content and may require basic website design skills and software.

Advantages

- Static Websites are easier to develop
- Can be developed quickly
- They are indexed easily by search engines as all the web pages actually exist on the server, which is not the case with dynamic websites.

Disadvantages

- Static websites cannot do complex tasks required by many online services.
- Updating a whole site can be cumbersome and time consuming.
- An isolation of Data and Design is not provided in static websites.

Dynamic Website

A dynamic website is one that changes or customizes itself frequently and automatically, based on certain criteria.

Dynamic websites can have two types of dynamic activity: Code and Content. Dynamic code is invisible or behind the scenes and dynamic content is visible or fully displayed

Advantages

- Can do more complex task required by online services.
- They are easier to update.
- Isolation of data and design allows programmers and content writers to work independently.

Disadvantages

- Can take more time to build.
- Can be difficult to build.
- Dynamic websites are not indexed by search engines easily, since they do not have actual web pages present on the web server. With continuous improvements in search engine technology, this problem is now very much eliminated and you can find that many dynamic websites are very well indexed by search engines now a days.

A few such classifications are:

- Affiliate: enabled portal that renders not only its custom CMS but also syndicated content from other content providers for an agreed fee. There are usually three relationship tiers. Affiliate Agencies (e.g., Commission Junction), Advertisers (e.g., eBay) and consumer (e.g., Yahoo!).
- Archive site: used to preserve valuable electronic contents that are on verge of extinction. For examples: Internet Archive, which since 1996 has preserved billions of old and new web pages; and Google Groups, which in early 2005 had preserved over 845,000,000 messages posted to Usenet news/discussion groups.
- Blog Site: sites generally used to post online diaries, comments or views that may include discussion forums (e.g., blogger, Xanga).
- Content Site: these sites create and sell of original content to end-user. (e.g., Slate, About.com).
- Corporate website: used to provide information regarding business, organization, or service.
- Commerce site (or eCommerce site): these sites are designed for purchasing or selling goods, such as Amazon.com, CSN Stores, and Overstock.com.
- Community site: sites where persons with similar interests communicate to each other through chatting and messaging or through soci message boards, such as MySpace or Facebook.
- City Site: A site that shows information about a certain city or town and events that takes place in that town. Usually created by the city council. For example, Richmond.com is the geodomain for Richmond, Virginia.

- Information site: contains content that is intended to inform visitors, but not necessarily for commercial purposes, such as: RateMyProfessors.com, Free Internet Lexicon and Encyclopaedia. Most government, educational and non-profit institutions have an informational site.
- Mirror site: A complete reproduction of a website.
- News site: similar to an information site, but dedicated to dispensing news and commentary.
- Personal homepage: run by an individual or a small group such as a family that contains information or any content that the individual wishes to include. These are usually uploaded using a web hosting service such as Geocities.

- Phish Site: a website created to fraudulently acquire sensitive information, such as passwords and credit card details, by disguising as a trustworthy person or business (such as Social Security Administration, PayPal) in an electronic communication.
- Political site: A site on which people may voice political views.
- Rating site: A site on which people can praise or disparage what is featured.
- Review site: A site on which people can post reviews for products or services.
- School site: a site on which teachers, students, or administrators can post information about current events at or involving their school.
- Video sharing: A site that enables user to upload videos, such as YouTube and Google Video.
- Search engine site: a site that provides general information and is intended as a gateway for retrieving other sites. Google, Yahoo and MSN are the most widely known search engines.
- Shock site: includes images or other material that is intended to be offensive to most viewers (e.g. rotten.com).
- Warez: a site designed to host and let users download copyrighted materials illegally.
- Web portal: a site is vehicle that provides a gateway to other resources on the Internet or an intranet.

Web Pages

A web page or webpage is a Document or information resource that is suitable for the world wide Web and can be accessed through a web browser and displayed on a computer display or mobile device. This information is usually in HTML or XHTML format, and may provide navigation bar to other web pages via Hyper text Hyper link. Web pages frequently subsume other resources such as Cascading Style Sheet, Client-side-scripting and Images into their final presentation.

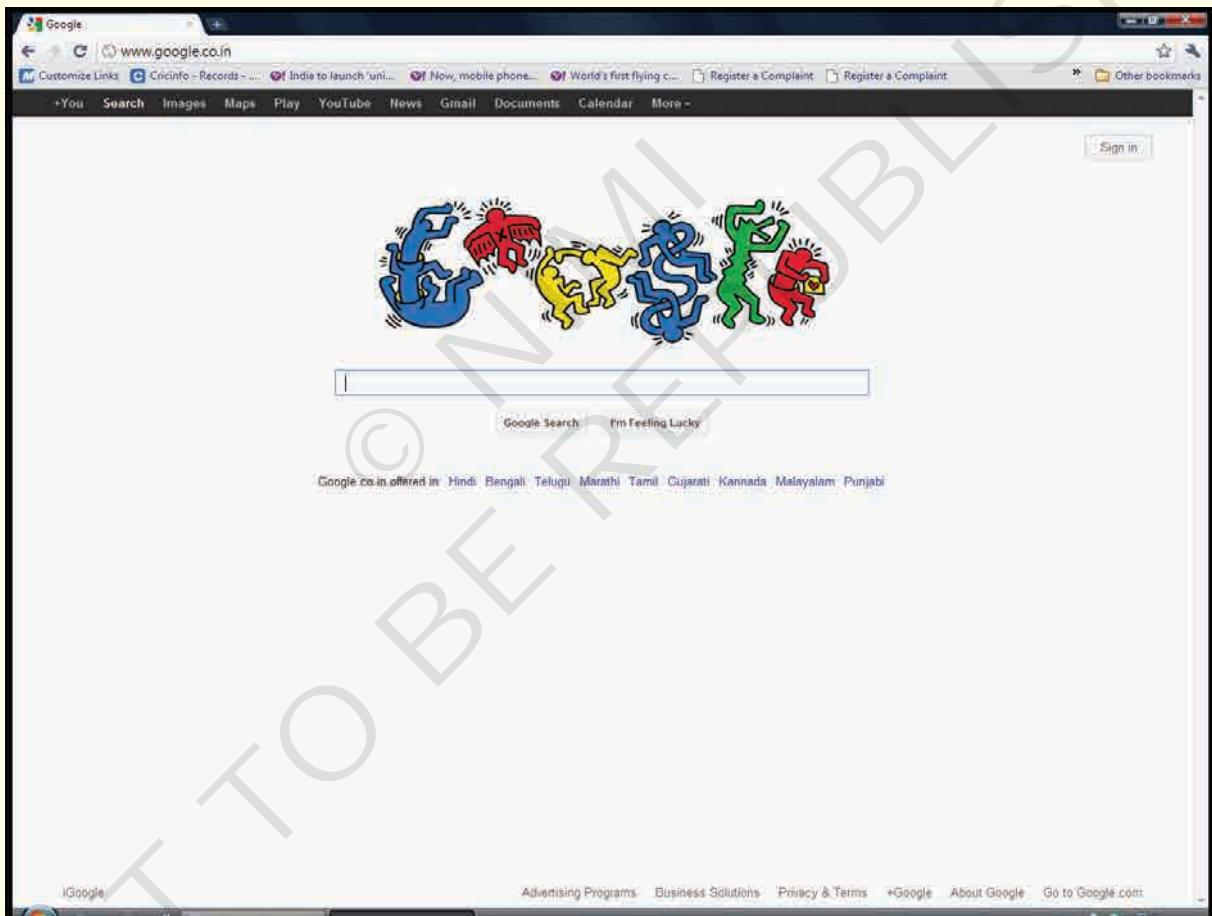
Web pages may be retrieved from a local computer or from a remote Web server. The web server may restrict access only to a private network, e.g. a corporate Intranet or it may publish pages on the World Wide Web. Web pages are requested and served from web. Web server using Hypertext Transfer Protocol (HTTP).

Web pages may consist of files of static text and other Web content stored within the Web server's file system(Static Web page), or may be constructed by Server-side scripting when they are requested (Dynamic web page). Client-side scripting can make web pages more responsive to user input once on the client browser.

Web Browser

A Web browser can have a Graphical User Interface, like Internet Explorer, Mozilla Firefox, Google Chrome and Opera (web browser), or can be Command Line Interface, like Lynx (web browser) (Fig 1) or Links (web browser).

Fig 1



Downloading a Software from Internet

Web users with disabilities often use assistive technologies and adaptive strategies to Web accessibility web pages. Users may be colour blind, may or may not want to use a mouse perhaps due to repetitive stress injury or motor-neurone problems, may be deaf and require audio to be captioned, may be blind and using a Screen reader or display, may need screen magnification, etc. Disabled and able-bodied users may disable the download and viewing of images and other media, to save time, network bandwidth or merely to simplify their browsing experience.

Users of mobile devices often have restricted displays and bandwidth. Anyone may prefer not to use the fonts, font sizes, styles and colour schemes selected by the web page designer and may apply their own CSS styling to the page. The World Wide Web Consortium (W3C) and Web Accessibility Initiative (WAI) recommend that all web pages should be designed with all of these options in mind.

- 1 Create a Temporary Files folder by opening My Computer, double click on your hard drive (typically the C: drive), then select File/New/Folder as on Fig 2.

- 2 Type "Temporary File" and name it as on Fig 3.
- 3 Type "My SQL 5. 1 free download " from freeware software website on internet.

Fig 2

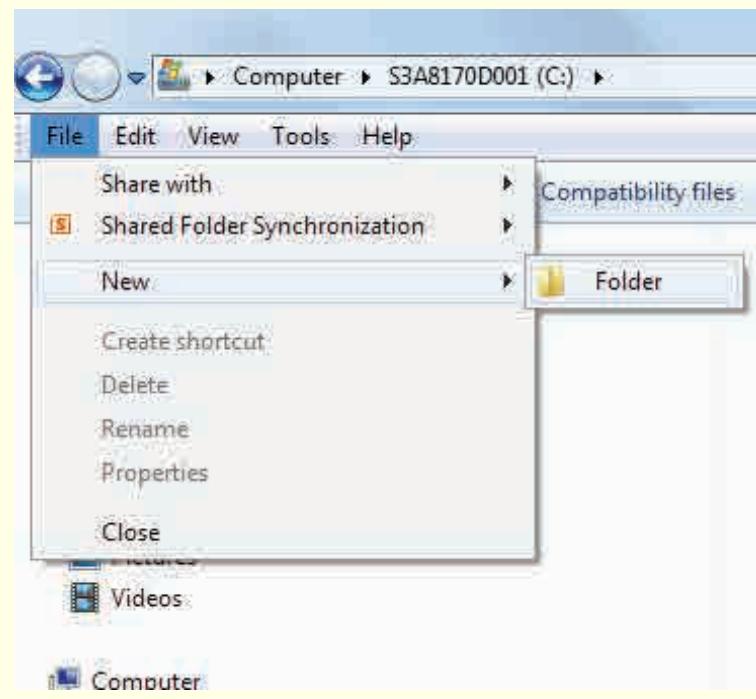
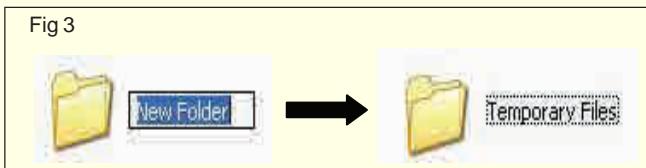
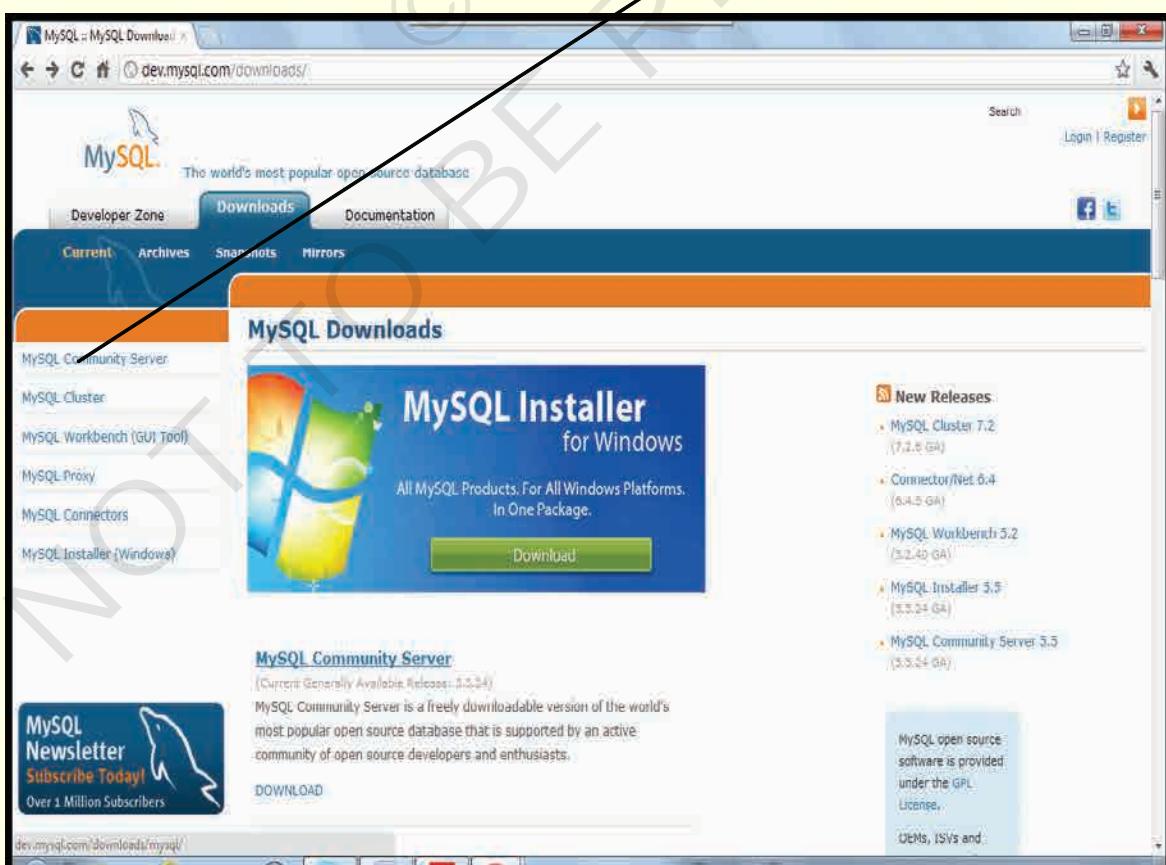


Fig 3



Note: As an example Choose "http://dev.mysql.com/downloads/" select "MySQL Community server" in download from the opened site as on Fig 4

Fig 4



- 4 Click "MySQL Community Server 5.1" from looking for previous version option as on Fig 5

Fig 5

Download MySQL Community Server

MySQL Community Edition is a freely downloadable version of the world's most popular open source database that is supported by an active community of open source developers and enthusiasts.

MySQL Cluster Community Edition is available as a separate download. The reason for this change is so that MySQL Cluster can provide more frequent updates and support using the latest sources of MySQL Cluster Carrier Grade Edition.

Important Platform Support Updates

Online Documentation

- MySQL 5.5 Documentation and Change History for the Generally Available (GA) Release
- MySQL 5.1 Documentation and Change History for the previous Generally Available (GA) Release

Looking for previous GA versions?

- MySQL Community Server 5.1 »
- MySQL Community Server 5.0 »

- 5 Choose "Windows (x86, 32-bit), MSI Installer Essentials - Recommended" and click download. (Fig 6)

Fig 6

Generally Available (GA) Releases Development Releases

MySQL Community Server 5.1.63

Select Version:

5.1.63 Select

Select Platform:

Microsoft Windows Select

Platform	Version	File Size	Action
Windows (x86, 32-bit), MSI Installer Essentials - Recommended	5.1.63	38.7M	Download
Windows (x86, 64-bit), MSI Installer Essentials - Recommended	5.1.63	31.4M	Download
Windows (x86, 64-bit), MSI Installer	5.1.63	99.0M	Download
Windows (x86, 32-bit), MSI Installer	5.1.63	105.9M	Download

Note: Save the "MySQL 5.1.63" in the created Folder name "Temporary Folder"

- 6 Burn the Downloaded "MySQL 5.1.63" in a CD ROM for Installation.

WEB LANGUAGES

Web languages are called as Markup languages are designed for the processing, definition and presentation of text. The language specifies code for formatting, both the layout and style, within a text file. The code used to specify the formatting are called tags

Four Types of Markup languages

- 1 BML
- 2 HTML
- 3 DHTML
- 4 XML

BML (Better markup language): BML is essentially a simple macro language. Macros are called blocks in BML. Blocks are defined in look files, and are invoked in BML files. Blocks accept parameters and are divided into several types, according to how parameters are transmitted and how the definition of the block is able to make use of them.

HTML (Hyper text markup Language): HTML or HyperText Markup Language is the language of the web. All web pages are written in HTML. HTML defines the way that images, multimedia, and text are displayed in web browsers. It includes elements to connect the documents (hypertext) and make web documents interactive (such as with forms).

HTML is a defined standard markup language. That standard was developed by the World Wide Web Consortium (W3C). It is based upon SGML (Standard Generalized Markup Language). It is a language that uses tags to define the structure of your text. Elements and tags are defined by the < and > characters.

DHTML: Dynamic HTML is not really a new specification of HTML, but rather a new way of looking at and controlling the standard HTML codes and commands.

When thinking of dynamic HTML, we need to remember the qualities of standard HTML, especially that once a page is loaded from the server, it will not change until another request comes to the server. Dynamic HTML give more control over the HTML elements and allows them to change at any time, without returning to the Web server.

There are four parts to DHTML:

- Document Object Model (DOM) (definition)
- Scripts
- Cascading Style Sheets (CSS)
- XHTML

DOM: The DOM is allows to access any part of Web page to change it with DHTML. Every part of a Web page is specified by the DOM and using its consistent naming conventions can access them and change their properties.

Scripts: Scripts written in either JavaScript or ActiveX are the two most common scripting languages used to activate DHTML. You use a scripting language to control the objects specified in the DOM.

Cascading Style Sheets: CSS is used in DHTML to control the look and feel of the Web page. Style sheets define the colors and fonts of text, the background colors and images, and the placement of objects on the page. Using scripting and the DOM, we can change the style of various elements.

XHTML: XHTML or HTML 4.x is used to create the page itself and build the elements for the CSS and the DOM to work on. There is nothing special about XHTML for DHTML - but having valid XHTML is even more important, as there are more things working from it than just the browser.

Features of DHTML

There are four primary features of DHTML:

- 1 Changing the tags and properties
- 2 Real-time positioning
- 3 Dynamic fonts (Netscape Communicator)
- 4 Data binding (Internet Explorer)

Changing the tags and properties

This is one of the most common uses of DHTML. It allows to change the qualities of an HTML tag depending on an event outside of the browser (such as a mouse click, time, or date, and so on). we can use this to preload information onto a page, and not display it unless the reader clicks on a specific link.

Real-time positioning

Objects, images, and text moving around the Web page. This can allow we to play interactive games with the readers or animate portions of the screen.

Dynamic Fonts

This is a Netscape only feature. Netscape developed this to get around the problem designers had with not knowing what fonts would be on a reader's system. With dynamic fonts, the fonts are encoded and downloaded with the page, so that the page always looks how the designer intended it to.

Data binding: This is an IE only feature. Microsoft developed this to allow easier access to databases from Web sites. It is very similar to using a CGI to access a database, but uses an ActiveX control to function.

XML: Extensible Markup Language (XML) is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable. It is defined in the XML 1.0 Specification produced by the W3C, and several other related specifications, all gratis open standards. The design goals of XML emphasize simplicity, generality, and usability over the Internet. It is a textual data format with strong support via Unicode for the languages of the world. Although the design of XML focuses on documents, it is widely used for the representation of arbitrary data structures, for example in web services

Creating an HTML document: Before start writing code to write a web page, it is a good practice to plan ahead the appearance of the web page. An HTML document has two elements:

- 1 Document Content
- 2 Tags

Document content is the information on a web page that the user will see. That information could be text or graphics.

Tags are the HTML codes that control how the document content will appear. The tags, in other words, will determine whether the text will be bold, black or blue, or of font type Time New Roman or Arial.

Start Notepad

To start Notepad go to:

Start

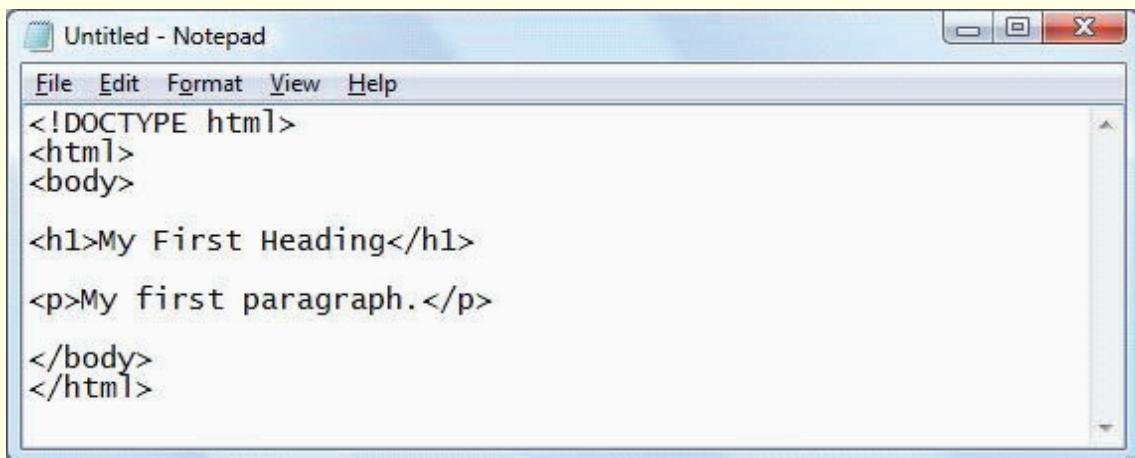
All Programs

Accessories

Notepad

Edit Your HTML with Notepad (Fig 7)

Fig 7



A screenshot of a Windows Notepad window titled "Untitled - Notepad". The window contains the following HTML code:

```
<!DOCTYPE html>
<html>
<body>

<h1>My First Heading</h1>

<p>My first paragraph.</p>

</body>
</html>
```

Type your HTML code into your Notepad:

Save Your HTML

Select Save as.. in Notepad's file menu.

When you save an HTML file, you can use either the .htm or the .html file extension.

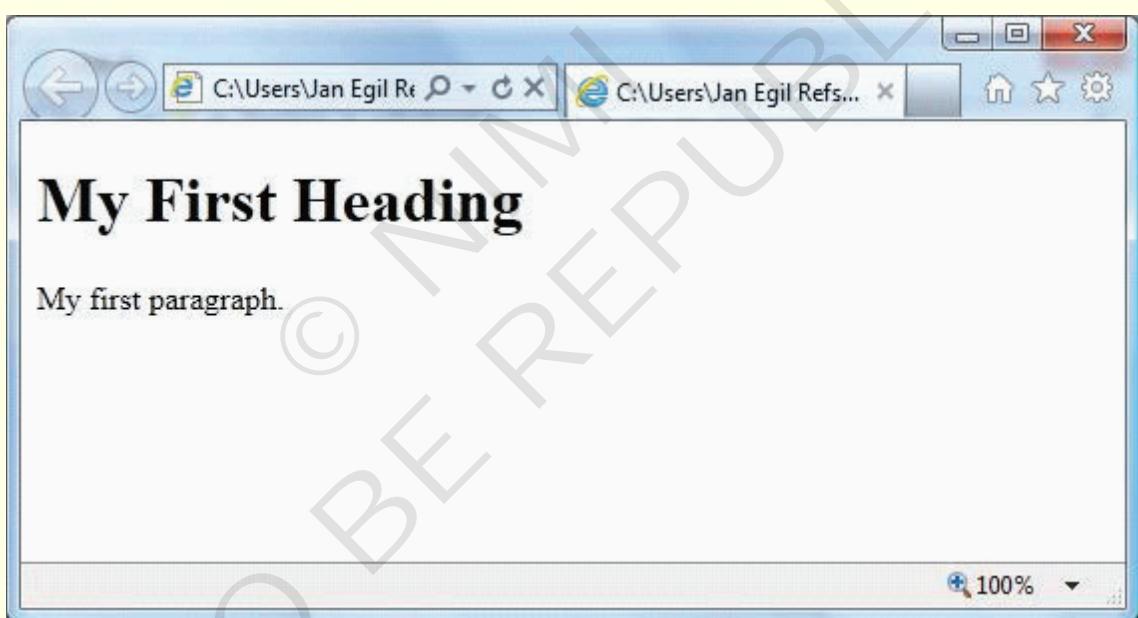
Save the file in a folder that is easy to remember

Run the HTML in Your Browser

Start your web browser and open your html file from the File, Open menu, or just browse the folder and double-click your HTML file.

The result should look much like this: (Fig 8)

Fig 8



Structure of Markup Language

An HTML document has two* main parts:

- 1 head. The head element contains title and meta data of a web document.
- 2 body. The body element contains the information that you want to display on a web page.

To make your web pages compatible with HTML 4, you need to add a document type declaration (DTD) before the HTML element. Many web authoring software add DTD and basic tags automatically when you create a new web page.

In a web page, the first tag (specifically, <html>) indicates the markup language that is being used for the document. The <head> tag contains information about the web page. Lastly, the content appears in the <body> tag. (Fig 9)

The <!DOCTYPE> Declaration

There are many different documents on the web. A browser can only display a document correctly, if it knows what kind of document it is.

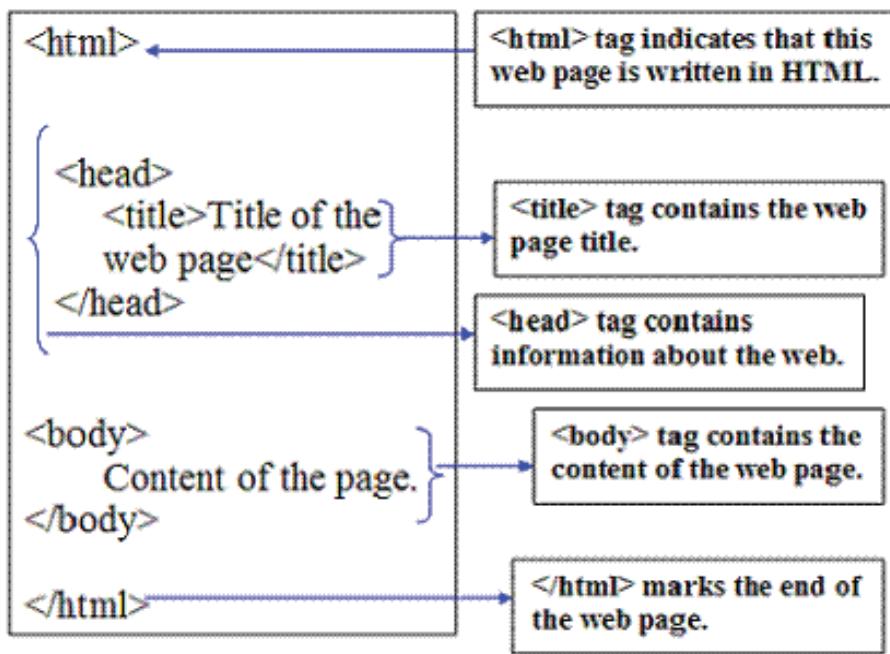
There are also many different versions of HTML, and a browser can only display an HTML page 100% correctly if it knows the exact HTML version used in the page. This is what <!DOCTYPE> is used for.

<!DOCTYPE> is not an HTML tag. It is an information (a declaration) to the browser about what version the HTML is written in.

The HTML <head> Element

The <head> element is a container for all the head elements. Elements inside <head> can include scripts, instruct the browser where to find style sheets, provide meta information, and more.

Fig 9



The following tags can be added to the head section: **<title>**, **<base>**, **<link>**, **<meta>**, **<script>**, and **<style>**.

The HTML **<title>** Element

The **<title>** tag defines the title of the document.

The title element is required in all HTML/XHTML documents.

The title element:

- Defines a title in the browser toolbar.
- Provides a title for the page when it is added to favorites.
- Displays a title for the page in search-engine results.

HTML Element Syntax

- An HTML element starts with a start tag / opening tag
- An HTML element ends with an end tag / closing tag
- The element content is everything between the start and the end tag
- Some HTML elements have empty content
- Empty elements are closed in the start tag
- Most HTML elements can have attributes

HTML Headings

HTML headings are defined with the **<h1>** to **<h6>** tags.

Examples

```
<html>
<body>
<h1>This is heading 1</h1>
<h2>This is heading 2</h2>
<h3>This is heading 3</h3>
```

<h4>This is heading 4</h4>

<h5>This is heading 5</h5>

<h6>This is heading 6</h6>

</body>

</html>

Result

This is heading 1

This is heading 2

This is heading 3

This is heading 4

This is heading 5

This is heading 6

HTML Paragraphs

HTML paragraphs are defined with the **<p>** tag.

<html>

<body>

<p>This is a paragraph.</p>

<p>This is a paragraph.</p>

<p>This is a paragraph.</p>

</body>

</html>

Examples

This is a paragraph.

This is a paragraph.

This is a paragraph.

HTML Links

HTML links are defined with the `<a>` tag.

```
<html>
<body>
<a href="http://www.facebook.com">
This is a link</a>
</body>
</html>
```

Result

This is a link

By clicking the link it shows the facebook login page

HTML Images

HTML images are defined with the `` tag.

```
<html>
<body>

</body>
</html>
```

Result (Fig 10)



Fig 10

HTML Attributes

- HTML elements can have attributes
- Attributes provide additional information about an element
- Attributes are always specified in the start tag
- Attributes come in name/value pairs like: `name="value"`

Attribute Example

HTML links are defined with the `<a>` tag. The link address is specified in the `href` attribute:

```
<html>
<body>
<a href="http://www.yahoo.com">
This is a link</a>
</body>
</html>
```

Result

This is the link

By clicking the link yahoo home page appears (Fig 11).

Formatting

Create Line Breaks - The `
` Element:

Whenever the `
` element, anything following it starts on the next line. This tag is an example of an empty element, where you do not need opening and closing tags, as there is nothing to go in between them.



Example:

```
Hello<br/>  
You come most carefully upon your hour.<br/>  
Thanks<br/>  
Mahnaz
```

Result

```
Hello  
You come most carefully upon your hour.  
Thanks  
Mahnaz  
To Become
```

Centring Content - The <center> Element:

You can use <center> tag to put any content in the center of the page or any table cell.

Example

```
<p>This is not in the center.</p>  
<center>  
<p>This is in the center.</p>  
</center>
```

This will produce following result:

```
This is not in the center.  
This is in the center.
```

Soft Hyphens

Occasionally, you will want to allow a browser to hyphenate long words to better justify a paragraph. For example, consider the following code and its resulting output.

```
<p style="text-align: justify;"> The morbid fear of the number 13, or triskaidekaphobia, has plagued some important historic figures like Mahamiya and Nanao.</p>
```

This will produce following result

Example for soft hyphen - The morbid fear of the number 13, or triskaidekaphobia, has plagued some important historic figures like Mahamiya and Nanao.

Preserve Formatting - The <pre> Element

Sometimes you want your text to follow the exact format of how it is written in the HTML document. In those cases, you can use the preformatted tag (<pre>).

Any text between the opening <pre> tag and the closing </pre> tag will preserve the formatting of the source document.

```
<pre>  
function testFunction( strText ){  
alert (strText)  
}  
</pre>
```

This will produce following result:

```
function testFunction( strText ){  
alert (strText)  
}
```

Horizontal Rules - The <hr /> Element

Horizontal rules are used to visually break up sections of a document. The <hr> tag creates a line from the current position in the document to the right margin and breaks the line accordingly.

For example you may want to give a line between two paragraphs as follows:

```
<p>This is paragraph one and should be on top</p>  
<hr />  
<p>This is paragraph two and should be at bottom</p>
```

This will produce following result:

```
This is paragraph one and should be on top  
This is paragraph two and should be at bottom
```

Again <hr /> tag is an example of an empty element, where you do not need opening and closing tags, as there is nothing to go in between them.

Note: The <hr /> element has a space between the characters hr and the forward slash. If you omit this space, older browsers will have trouble rendering the line break, while if you miss the forward slash character and just use <hr> it is not valid XHTML

Presentational Tags

If you use a word processor, you are familiar with the ability to make text bold, italicized, or underlined; these are just three of the ten options available to indicate how text can appear in HTML and XHTML.

Bold Text - The Element

Anything that appears in a ... element is displayed in bold, like the word bold here:

```
<p>The following word uses a <b>bold</b> typeface.</p>
```

This will produce following result:

The following word uses a bold typeface.

Italic Text - The <i> Element

Anything that appears in a <i>...</i> element is displayed in italicized, like the word italicized here:

```
<p>The following word uses a <i>italicized</i> typeface.</p>
```

This will produce following result:

The following word uses a italicized typeface.

Underlined Text - The <u> Element: Anything that appears in a <u>...</u> element is displayed with underline, like the word underlined here:

Related Theory for Exercise 1.31.113

```
<p>The following word uses a <u>underlined</u> typeface.</p>
```

This will produce following result:

The following word uses a underlined typeface.

Strike Text - The **<strike>** Element:

Anything that appears in a **<strike>...</strike>** element is displayed with strikethrough, which is a thin line through the text:

```
<p>The following word uses a <strike>strikethrough</strike> typeface.</p>
```

This will produce following result:

The following word uses a strikethrough typeface.

Monospaced font - The **<tt>** Element:

The content of a **<tt>** element is written in monospaced font. Most fonts are known as variable-width fonts because different letters are of different widths (for example, the letter m is wider than the letter i). In a monospaced font, however, each letter is the same width.

```
<p>The following word uses a <tt>monospaced</tt> typeface.</p>
```

This will produce following result:

The following word uses a monospaced typeface.

Superscript Text - The **<sup>** Element:

The content of a **<sup>** element is written in superscript; the font size used is the same size as the characters surrounding it but is displayed half a character's height above the other characters.

```
<p>The following word uses a <sup>superscript</sup> typeface.</p>
```

This will produce following result:

The following word uses a superscript typeface.

Subscript Text - The **<sub>** Element:

The content of a **<sub>** element is written in subscript; the font size used is the same as the characters surrounding it, but is displayed half a character's height beneath the other characters.

```
<p>The following word uses a <sub>subscript</sub> typeface.</p>
```

This will produce following result:

The following word uses a subscript typeface.

Larger Text - The **<big>** Element:

The content of the **<big>** element is displayed one font size larger than the rest of the text surrounding it.

```
<p>The following word uses a <big>big</big> typeface.</p>
```

This will produce following result:

The following word uses a big typeface.

Smaller Text - The **<small>** Element:

The content of the **<small>** element is displayed one font size smaller than the rest of the text surrounding it.

```
<p>The following word uses a <small>small</small> typeface.</p>
```

This will produce following result:

The following word uses a small typeface.

Styling HTML with CSS

CSS was introduced together with HTML 4, to provide a better way to style HTML elements.

CSS can be added to HTML in the following ways:

- Inline - using the style attribute in HTML elements
- Internal - using the **<style>** element in the **<head>** section
- External - using an external CSS file

```
<html>
```

```
<body style="background-color:PowderBlue;">
```

```
<h1>Look! Styles and colors</h1>
```

```
<p style="font-family:verdana;color:red;">
```

This text is in Verdana and red</p>

```
<p style="font-family:times;color:green;">
```

This text is in Times and green</p>

```
<p style="font-size:30px;">This text is 30 pixels high</p>
```

```
</body>
```

```
</html>
```

Result:

Look! Styles and colors

This text is in Verdana and red

This text is in Times and green

This text is 30 pixels high

HTML Hyperlinks (Links)

A hyperlink (or link) is a word, group of words, or image that you can click on to jump to a new document or a new section within the current document.

When you move the cursor over a link in a Web page, the arrow will turn into a little hand.

Links are specified in HTML using the **<a>** tag.

The **<a>** tag can be used in two ways:

- 1 To create a link to another document, by using the href attribute
- 2 To create a bookmark inside a document, by using the name attribute

The HTML code for a link is simple. It looks like this:

```
<a href="url">Link text</a>
```

The href attribute specifies the destination of a link.

Visit yahoo

which will display like this: Visit yahoo.com

Clicking on this hyperlink will send the user to Yahoo homepage.

The "Link text" doesn't have to be text. It can be an image or any other HTML element.

HTML Links - The target Attribute

The target attribute specifies where to open the linked document.

The example below will open the linked document in a new browser window or a new tab:

Example

```
<a href="http://www.yahoo.com/" target="_blank">Visit yahoo !</a>
```

```
<html>
```

```
<body>
```

```
<a href="http://www.yahoo.com" target="_blank">Visit yahoo.com!</a>
```

<p>If you set the target attribute to "_blank", the link will open in a new browser window/tab.</p>

```
</body>
```

```
</html>
```

Result

Visit yahoo.com!

If you set the target attribute to "_blank", the link will open in a new browser window/tab.

HTML Images - The Tag and the Src Attribute

In HTML, images are defined with the tag.

The tag is empty, which means that it contains attributes only, and has no closing tag.

To display an image on a page, you need to use the src attribute. Src stands for "source". The value of the src attribute is the URL of the image you want to display.

Syntax for defining an image:

```

```

The URL points to the location where the image is stored. An image named "bamboo.gif", located in the "images" directory on "www.w3schools.com" has the URL: http://www.backgroundlabs.com/index.php?search=bamboo.

The browser displays the image where the tag occurs in the document. If you put an image tag between two paragraphs, the browser shows the first paragraph, then the image, and then the second paragraph.

HTML Images - The Alt Attribute

The required alt attribute specifies an alternate text for an image, if the image cannot be displayed.

The value of the alt attribute is an author-defined text:

```

```

The alt attribute provides alternative information for an image if a user for some reason cannot view it (because of slow connection, an error in the src attribute, or if the user uses a screen reader).

HTML Images - Set Height and Width of an Image

The height and width attributes are used to specify the height and width of an image.

The attribute values are specified in pixels by default:

```

```

Note: It is a good practice to specify both the height and width attributes for an image. If these attributes are set, the space required for the image is reserved when the page is loaded. However, without these attributes, the browser does not know the size of the image. The effect will be that the page layout will change during loading (while the images load).

If an HTML file contains ten images - eleven files are required to display the page right. Loading images takes time, so my best advice is: Use images carefully.

When a web page is loaded, it is the browser, at that moment, that actually gets the image from a web server and inserts it into the page. (Fig 12) Therefore, make sure that the images actually stay in the same spot in relation to the web page, otherwise your visitors will get a broken link icon. The broken link icon is shown if the browser cannot find the image.

```
<html>
```

```
<body>
```

```
<h2>Friendship Card</h2>
```

```

```

```
</body>
```

```
</html>
```

HTML Tables

Tables are defined with the <table> tag.

A table is divided into rows (with the <tr> tag), and each row is divided into data cells (with the <td> tag). td stands for "table data," and holds the content of a data cell. A <td> tag can contain text, links, images, lists, forms, other tables, etc.

Table Example

```
<table border="1">
<tr>
<td>row 1, cell 1</td>
<td>row 1, cell 2</td>
</tr>
```

Fig 12



```
<tr>
<td>row 2, cell 1</td>
<td>row 2, cell 2</td>
</tr>
</table>
```

How the HTML code above looks in a browser:

row 1, cell 1 row 1, cell 2
row 2, cell 1 row 2, cell 2

HTML Tables and the Border Attribute

If you do not specify a border attribute, the table will be displayed without borders. Sometimes this can be useful, but most of the time, we want the borders to show.

To display a table with borders, specify the border attribute:

```
<table border="1">
<tr>
<td>Row 1, cell 1</td>
<td>Row 1, cell 2</td>
</tr>
</table>
```

HTML Table Headers

Header information in a table are defined with the `<th>` tag.

All major browsers display the text in the `<th>` element as bold and centered.

```
<table border="1">
<tr>
<th>Header 1</th>
<th>Header 2</th>
</tr>
<tr>
<td>row 1, cell 1</td>
<td>row 1, cell 2</td>
</tr>
<tr>
<td>row 2, cell 1</td>
<td>row 2, cell 2</td>
</tr>
</table>
```

How the HTML code above looks in your browser:

Header 1	Header 2
row 1, cell 1	row 1, cell 2
row 2, cell 1	row 2, cell 2

HTML Unordered Lists

An unordered list starts with the `` tag. Each list item starts with the `` tag.

The list items are marked with bullets (typically small black circles).

```
<ul>
<li>Coffee</li>
<li>Milk</li>
</ul>
```

How the HTML code above looks in a browser:

- Coffee
- Milk

HTML Ordered Lists

An ordered list starts with the `` tag. Each list item starts with the `` tag.

The list items are marked with numbers.

```
<ol>
<li>Coffee</li>
<li>Milk</li>
</ol>
```

How the HTML code above looks in a browser:

- 1 Coffee
- 2 Milk

HTML Definition Lists

A definition list is a list of items, with a description of each item.

The `<dl>` tag defines a definition list.

The `<dl>` tag is used in conjunction with `<dt>` (defines the item in the list) and `<dd>` (describes the item in the list):

```
<dl>
<dt>Coffee</dt>
<dd>- black hot drink</dd>
<dt>Milk</dt>
<dd>- white cold drink</dd>
</dl>
```

How the HTML code above looks in a browser:

Coffee - black hot drink

Milk- white cold drink

Note : Inside a list item you can put text, line breaks, images, links, other lists, etc.

HTML List Tags

Tag	Description
<code></code>	Defines an ordered list
<code></code>	Defines an unordered list
<code></code>	Defines a list item
<code><dl></code>	Defines a definition list
<code><dt></code>	Defines an item in a definition list
<code><dd></code>	Defines a description of an item in a definition list

HTML elements can be grouped together with `<div>` and ``

HTML Block Elements

Most HTML elements are defined as block level elements or as inline elements.

Block level elements normally start (and end) with a new line when displayed in a browser.

Examples: `<h1>`, `<p>`, ``, `<table>`

HTML Inline Elements

Inline elements are normally displayed without starting a new line.

Examples: ``, `<td>`, `<a>`, ``

The HTML `<div>` Element

The HTML `<div>` element is a block level element that can be used as a container for grouping other HTML elements.

The `<div>` element has no special meaning. Except that, because it is a block level element, the browser will display a line break before and after it.

When used together with CSS, the `<div>` element can be used to set style attributes to large blocks of content.

Another common use of the `<div>` element, is for document layout. It replaces the "old way" of defining layout using tables. Using tables is not the correct use of the `<table>` element. The purpose of the `<table>` element is to display tabular data.

The HTML `` Element

The HTML `` element is an inline element that can be used as a container for text.

The `` element has no special meaning.

When used together with CSS, the `` element can be used to set style attributes to parts of the text.

HTML Grouping Tags

Tag	Description
<code><div></code>	Defines a div
<code></code>	Defines a span

HTML Layout

Website Layouts

Most websites have put their content in multiple columns (formatted like a magazine or newspaper).

Multiple columns are created by using `<div>` or `<table>` elements. CSS are used to position elements, or to create backgrounds or colorful look for the pages.

HTML Layouts - Using `<div>` Elements

The `div` element is a block level element used for grouping HTML elements.

The following example uses five `div` elements to create a multiple column layout, creating the same result as in the previous example:

Example

```
<div>
<!DOCTYPE html>
<html>
<body>
<div id="container" style="width:500px">
<div id="header" style="background-color:#FFA500;">
<h1 style="margin-bottom:0;">Main Title of Web Page</h1></div>
<div id="menu" style="background-color:#FFD700;float:left;">
<b>Menu</b><br />
```

HTML

CSS

JavaScript</div>

```
<div id="content" style="background-color:#EEEEEE;height:200px;width:400px;float:left;">
Content goes here</div>
<div id="footer" style="background-color:#FFA500;clear:both;text-align:center;">
</div>
</div>
</body>
</html>
```

Result (Fig 13)

Fig 13

Main Title of Web Page

Menu

HTML

CSS

JavaScript

Content goes here

HTML Forms

HTML forms are used to pass data to a server.

A form can contain input elements like text fields, checkboxes, radio-buttons, submit buttons and more. A form can also contain select lists, textarea, fieldset, legend, and label elements.

The <form> tag is used to create an HTML form:

<form>

input elements.

</form>

HTML Forms - The Input Element

The most important form element is the input element.

The input element is used to select user information.

An input element can vary in many ways, depending on the type attribute. An input element can be of type text field, checkbox, password, radio button, submit button, and more.

The most used input types are described below.

Text Fields

<input type="text" /> defines a one-line input field that a user can enter text into:

<form>

First name: <input type="text" name="firstname" />

Last name: <input type="text" name="lastname" />

</form>

How the HTML code above looks in a browser:

First name:

Last name:

Note: The form itself is not visible. Also note that the default width of a text field is 20 characters.

Password Field

<input type="password" /> defines a password field:

<form>

Password: <input type="password" name="pwd" />

</form>

How the HTML code above looks in a browser:

Password:

Note: The characters in a password field are mased (shown as asterisks or circles)

Radio Buttons

<input type="radio" /> defines a radio button. Radio buttons let a user select ONLY ONE of a limited number of choices:

```
<form>
<input type="radio" name="sex" value="male" /> Male<br/>
<input type="radio" name="sex" value="female" /> Female
</form>
```

How the HTML code above looks in a browser:

- Male
- Female

Checkboxes

<input type="checkbox" /> defines a checkbox. Checkboxes let a user select ONE or MORE options of a limited number of choices.

```
<form>
<input type="checkbox" name="vehicle" value="Bike" />
I have a bike<br />
<input type="checkbox" name="vehicle" value="Car" />
I have a car
</form>
```

How the HTML code above looks in a browser:

- I have a bike
- I have a car

Submit Button

<input type="submit" /> defines a submit button.

A submit button is used to send form data to a server. The data is sent to the page specified in the form's action attribute. The file defined in the action attribute usually does something with the received input:

```
<form name="input" action="html_form_action.asp"
method="get">
```

Username: <input type="text" name="user" />

<input type="submit" value="Submit" />

</form>

How the HTML code above looks in a browser:

Username: Submit

If you type some characters in the text field above, and click the "Submit" button, the browser will send your input to a page called "html_form_action.asp". The page will show you the received input.

HTML Iframes

Syntax for adding an iframe:

<iframe src="URL"></iframe>

The URL points to the location of the separate page.

Iframe - Set Height and Width

The height and width attributes are used to specify the height and width of the iframe.

The attribute values are specified in pixels by default, but they can also be in percent (like "80%").

Example

```
<!DOCTYPE html>
<html>
<body>
<iframe src="demo_iframe.htm" width="200"
height="200"></iframe>
</body>
</html>
```

It will appear as shown in Fig 14.

Iframe - Remove the Border

The frameborder attribute specifies whether or not to display a border around the iframe.

Set the attribute value to "0" to remove the border:

Example

```
<iframe src="demo_iframe.htm" frameborder="0"></
iframe>
```

Use iframe as a Target for a Link

An iframe can be used as the target frame for a link.

The target attribute of a link must refer to the name attribute of the iframe:

Example

```
<iframe src="demo_iframe.htm" name="iframe_a"></
iframe>
```

```
<p><a href = "http://www.yahoo.com"
target="iframe_a">yahoo.com</a></p>
```

HTML Iframe Tag

Tag	Description
<iframe>	Defines an inline sub window (frame)

HTML Colour

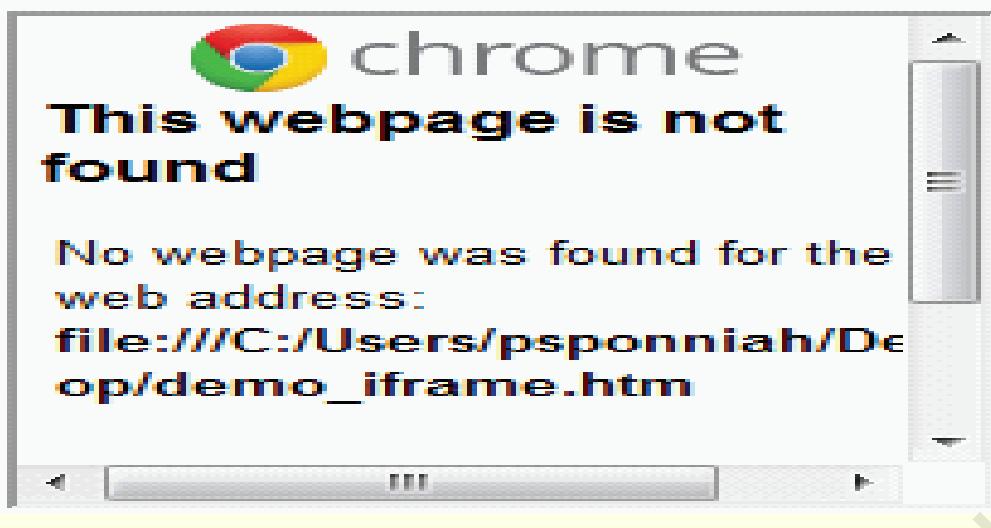
Color Values

HTML colors are defined using a hexadecimal notation (HEX) for the combination of Red, Green, and Blue color values (RGB).

The lowest value that can be given to one of the light sources is 0 (in HEX: 00). The highest value is 255 (in HEX: FF).

HEX values are specified as 3 pairs of two-digit numbers, starting with a # sign.

Fig 14



Color Values (Fig 15)

<!DOCTYPE html>	</p>
<html>	<p style="background-color:yellow">
<body>	Color set by using color name
<p style="background-color:#FFFF00">	</p>
Color set by using hex value	</body>
</p>	</html>
<p style="background-color:rgb(255,255,0)">	Result
Color set by using rgb value	Color set by using hex value
	Color set by using rgb value
	Color set by using color name

Fig 15

Color	Color HEX	Color RGB
Black	#000000	rgb(0,0,0)
Red	#FF0000	rgb(255,0,0)
Green	#00FF00	rgb(0,255,0)
Blue	#0000FF	rgb(0,0,255)
Yellow	#FFFF00	rgb(255,255,0)
Cyan	#00FFFF	rgb(0,255,255)
Magenta	#FF00FF	rgb(255,0,255)
Grey	#C0C0C0	rgb(192,192,192)
White	#FFFFFF	rgb(255,255,255)

DHTML

The HTML script Element

The <script> tag is used to define a client-side script, such as a JavaScript.

The script element either contains scripting statements or it points to an external script file through the src attribute.

The required type attribute specifies the MIME type of the script.

Common uses for JavaScript are image manipulation, form validation, and dynamic changes of content.

The script below writes Hello World! to the HTML output:

Example

```
<script type="text/javascript">  
document.write("Hello World!")  
</script>
```

The HTML noscript Element

The `<noscript>` tag is used to provide an alternate content for users that have disabled scripts in their browser or have a browser that doesn't support client-side scripting.

The noscript element can contain all the elements that you can find inside the body element of a normal HTML page.

The content inside the noscript element will only be displayed if scripts are not supported, or are disabled in the user's browser:

Example

```
<!DOCTYPE html>
<html>
<body>
<script type="text/javascript">
document.write("Hello World!")
</script>
<noscript>Sorry, your browser does not support
JavaScript!</noscript>
```

A browser without support for JavaScript will show the text in the noscript element.

```
</body>
</html>
```

Result

Hello World!

A browser without support for JavaScript will show the text in the noscript element.

HTML Script Tags

Tag	Description
<code><script></code>	Defines a client-side script
<code><noscript></code>	Defines an alternate content for users that do not support client-side scripts

HTML Entities

Some characters are reserved in HTML.

It is not possible to use the less than (`<`) or greater than (`>`) signs in your text, because the browser will mix them with tags.

To actually display reserved characters, we must use character entities in the HTML source code.

A character entity looks like this:

`&entity_name;`

OR

`&#entity_number;`

Non-breaking Space

A common character entity used in HTML is the non-breaking space (` `).

Browsers will always truncate spaces in HTML pages. If you write 10 spaces in your text, the browser will remove 9 of them, before displaying the page. To add spaces to your text, you can use the ` ` character entity.

HTML Useful Character Entities

Note: Entity names are case sensitive!

HTML Uniform Resource Locators

A **URL** is another word for a web address.

Result	Description	Entity Name	Entity Number
	non-breaking space	<code>&nbsp;</code>	<code>&#160;</code>
<code><</code>	less than	<code>&lt;</code>	<code>&#60;</code>
<code>></code>	greater than	<code>&gt;</code>	<code>&#62;</code>
<code>&</code>	ampersand	<code>&amp;</code>	<code>&#38;</code>
<code>¢</code>	cent	<code>&cent;</code>	<code>&#162;</code>
<code>£</code>	pound	<code>&pound;</code>	<code>&#163;</code>
<code>¥</code>	yen	<code>&yen;</code>	<code>&#165;</code>
<code>•</code>	euro	<code>&euro;</code>	<code>&#8364;</code>
<code>§</code>	section	<code>&sect;</code>	<code>&#167;</code>
<code>©</code>	copyright	<code>&copy;</code>	<code>&#169;</code>
<code>®</code>	registered trademark	<code>&reg;</code>	<code>&#174;</code>
<code>™</code>	trademark	<code>&trade;</code>	<code>&#8482;</code>

A URL can be composed of words, such as "w3schools.com", or an Internet Protocol (IP) address: 192.68.20.50. Most people enter the name of the website when surfing, because names are easier to remember than numbers.

URL - Uniform Resource Locator

When you click on a link in an HTML page, an underlying `<a>` tag points to an address on the world wide web.

A Uniform Resource Locator (URL) is used to address a document (or other data) on the world wide web.

Explanation

- **Scheme** - defines the **type** of Internet service. The most common type is http
- **Host** - defines the **domain host** (the default host for http is www)
- **Domain** - defines the Internet **domain name**, like w3schools.com
- **Port** - defines the **port number** at the host (the default port number for http is 80)

- **Path** - defines a **path** at the server (If omitted, the document must be stored at the root directory of the web site)

- **Filename** - defines the name of a document/resource

Common URL Schemes

The table below lists some common schemes:

Scheme	Short for....	Which pages will the scheme be used for...
http	HyperText Transfer Protocol	Common web pages starts with http://. Not encrypted
https	Secure HyperText Transfer Protocol	Secure web pages. All information exchanged are encrypted
ftp	File Transfer Protocol	For downloading or uploading files to a website. Useful for domain maintenance
file		A file on your computer

URL Encoding

URLs can only be sent over the Internet using the ASCII character-set.

Since URLs often contain characters outside the ASCII set, the URL has to be converted into a valid ASCII format.

URL encoding replaces non ASCII characters with a "%" followed by two hexadecimal digits.

URLs cannot contain spaces. URL encoding normally replaces a space with a + sign.

XML Structure

The XML structure including the document parts, the prologue, and provides a simple XML example document.

Document Parts

- Prolog
- Document Element (root element)

The Prologue

The prologue, equivalent to the header in HTML, may include the following:

- An XML declaration (optional) such as:

```
<?xml version="1.0"?>
```
- A DTD or reference to one (optional). An example reference to an external DTD file:

```
<!DOCTYPE LANGLIST SYSTEM "langlist.dtd">
```
- Processing instructions - An example processing instruction that causes style to be determined by a style sheet:

```
<?xml-stylesheet type="text/css" href="xmlstyle.css"?>
```

An XML Example

Therefore a complete well formed XML document may look like:

```
<?xml version="1.0"?>
<LAND>
  <FOREST>
    <TREE>Oak</TREE>
    <TREE>Pine</TREE>
    <TREE>Maple</TREE>
  </FOREST>
  <MEADOW>
    <GRASS>Bluegrass</GRASS>
    <GRASS>Fescue</GRASS>
    <GRASS>Rye</GRASS>
  </MEADOW>
</LAND>
```

The LAND element, above, is the root element.

Result

Oak Pine Maple Bluegrass Fescue Rye

Web Elements

A web page, as an information set, can contain numerous types of information, which is able to be seen, heard or interact by the End-user

Web Hosting

A web hosting service is a type of Internet hosting service that allows individuals and organizations to make their Website accessible via the World Wide Web. Web hosts are companies that provide space on a Server (computing) owned or leased for use by clients, as well as providing Internet connectivity, typically in a data centre. Web hosts can also provide data centre space and connectivity to the Internet for other servers located in their data centre, called Collocation.

Introduction to CMS and web authoring tools

Objectives: At the end of this lesson you shall be able to

- explain the meaning of content management System.
- list some of the popular CMSes.
- explain the use and main features of kompozer web authoring tool.

Definition of CMS

CMS Stands for "Content Management System." A CMS is a software tool that allows you to create, edit, and publish content.

Description of CMS

The goal of a CMS is to provide an intuitive user interface for building and modifying webpage content. Each CMS also provides a web publishing tool that allows one or more users to publish updates live on the Web. The editing component is called the content management application (CMA), while the publishing tool is called the content delivery application (CDA). These two components are integrated together in a CMS to streamline the web development process.

In terms of web publishing, content can be simple text, photos, music, video, documents, or just about anything you can think of. While early CMS software was used to manage documents and local computer files, most CMS systems are now designed exclusively to manage content on the Web. A major advantage of using a CMS is that it requires almost no technical skill or knowledge to manage.

Content management systems are available as installable applications and web-based user interfaces. The use of a web interfacesimplifies the website updating process. Additionally, most web-based CMSes are updated automatically, ensuring all users have the latest tools to manage their content.

A major advantage of using a CMS is that it requires almost no technical skill or knowledge to manage. Since the CMS manages all your content, you don't have to.

There are several web-based CMS tools available today. The following are some of the most popular ones:

- WordPress - free web software designed for creating template-based websites or blogs
- Blogger - Google's blogging tool designed specifically for maintaining a blog
- Joomla - a flexible web publishing tool that supports custom databases and extensions
- Drupal - an open source platform often used for developing community-based sites
- Weebly - a web-based platform for building simple personal and business websites

- Wix - a collection of web publishing tools for creating a highly customizable website

Introduction to WYSIWYG web authoring tools

WYSIWYG stands for "What You See Is What You Get". In such editors you edit not directly the source code of your documents, but its presentation as it will appear in the final document. So instead of writing blocks of code manually (as you e.g. would do it in Word or Notepad), you manipulate with design components using an editor window. This means that you view something very similar to the final result while the document or image is being created.

There are many easy-to-use WYSIWYG programs having all the tools needed to create a complex and fully functional websites, even by beginners. These sites even have options to work with HTML code be it design or editing. It is easier to create a Web site with an HTML editor, as software developers continue to add tools that let you develop advanced features with style. Many WYSIWYG web authoring tools offer advanced features to integrate Dynamic HTML or many other features into a site with an elegant and consistent design.

Introduction to Kompozer

KompoZer is a complete Web Authoring System that combines web file management and easy-to-use WYSIWYGweb page editing capabilities found in Microsoft FrontPage, Adobe DreamWeaver and other high end programs. With kompozer you can create web pages and manage a website with no technical expertise or knowledge of HTML.

A useful feature of Kompozer is that you can use it to publish your web pages to a web hosting server. You just need to know your FTP account username and password as well as the site URL and Kompozer will log in and place your web pages on the server.

KompoZer's File Menu contains menu items for all major operations possible while using it. KompoZer's tool bar user interface consists of:

- Composition Toolbar
- Format Toolbar
- Tab Browser Toolbar
- Edit Mode Toolbar
- Status Toolbar

Composition Toolbar

The Composition Toolbar consists of buttons for the most used major operations. Below is a snapshot of the default Composition Toolbar with text below each button indicating the respective button's function. (Fig 1)

Fig 1



Format Toolbar (Fig 2)

Fig 2



The Format Toolbar is a very useful tool while editing web pages with KompoZer. With the Format Toolbar you can apply paragraph format, choose a font, change foreground or background colour, increase or decrease size, and emphasize a block of text. You can also create ordered or unordered lists and justify a block of text to left, right or centre. To know which button does what just hover your mouse over the button and a tool tip will appear indicating the function of the button.

Tab Browser Toolbar (Fig 3)

Fig 3



KompoZer allows you to simultaneously edit multiple web documents using different tabs for each opened document. Having multiple tabs gives a cleaner look to your desktop as it is not cluttered when multiple windows are used for each document. As a visual indicator a "red floppy icon" icon appears for pages which have been edited but not saved yet.

Edit Mode Toolbar (Fig 4)

Fig 4



The Edit Mode Toolbar indicates the viewing mode which is presently active for the current document. Available viewing modes are the Normal view, HTML Tags view, HTML source view and the Preview mode. You can easily change your viewing mode by simply clicking any of the other three with the mouse.

Status Toolbar (Fig 5)

KompoZer's status bar shows the position of the cursor with respect to the HTML tag hierarchy. You can easily edit/assign the properties of any particular tag in the status bar just by right clicking and choosing the desired option. Simple left-click of the mouse on a tag in status bar selects the text surrounded by that tag.

Fig 5



Understanding JavaScript

Objectives : At the end of this lesson you shall be able to

- define programming and scripting languages
- know what is JavaScript and history of Java Script
- explain how to run JavaScript
- list out tools you need to run JavaScript
- view sample JavaScript Program
- know features of JavaScript
- describe advantages and disadvantages of JavaScript
- explain JavaScript Versions.

Introduction to programming and scripting languages

Computer **programming** is the process of writing instructions that get executed by computers. The instructions, also known as code, are written in a **programming** language which the computer can understand and use to perform a task or solve a problem.

A **script or scripting language** is a computer language with a series of commands within a file that is capable of being executed without being compiled. Good examples of server-side scripting languages include Perl, PHP, and Python. The best example of a client side scripting language is JavaScript.

Advantages of scripts

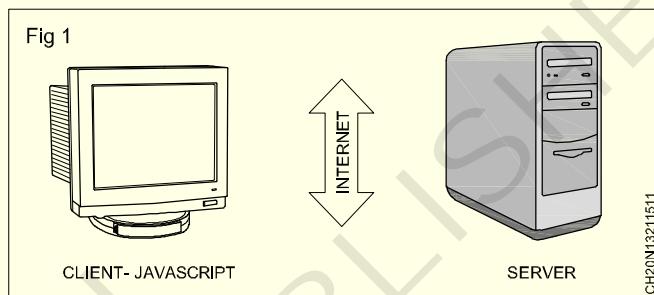
- Open source, allowing users to view and edit the script if needed.
- Does not require the file to be compiled, but may be when necessary.
- Easy to learn and write.
- Easy to port between different operating systems.
- Much faster to develop than an actual program - some individuals and companies write scripts as a prototype for actual programs.

Disadvantages of scripts

- Open source, allows others to view source code, which may be prohibited by some companies.
- Requires the user to install an interpreter or separate program before the script can be run.
- In some situations, they may be slower than a compiled program.

What is Java Script?

JavaScript is a very powerful **client-side scripting language**. JavaScript is used mainly for enhancing the interaction of a user with the webpage (Fig 1). In other words, you can make your webpage more lively and interactive, with the help of JavaScript. JavaScript is also being used widely in game development and Mobile application development.



JavaScript History

JavaScript was developed by Brendan Eich in 1995, which appeared in Netscape, a popular browser of that time. The language was initially called Live Script and was later renamed JavaScript. There are many programmers who think that JavaScript and Java are the same. In fact, **JavaScript and Java are very much unrelated. Java is a very complex programming language whereas JavaScript is only a scripting language.** The syntax of Java Script is mostly influenced by the programming language C.

How to Run JavaScript?

Being a scripting language, **JavaScript cannot run on its own. In fact, the browser is responsible for running JavaScript code.** When a user requests an HTML page with JavaScript in it, the script is sent to the browser and it is up to the browser to execute it. The main advantage of JavaScript is that **all modern web browsers support** JavaScript. So, you do not have to worry about whether your site visitor uses Internet Explorer, Google Chrome, Firefox or any other browser. JavaScript will be supported. Also, JavaScript **runs on any operating system** including Windows, Linux or Mac.

Tools You Need to run JavaScript

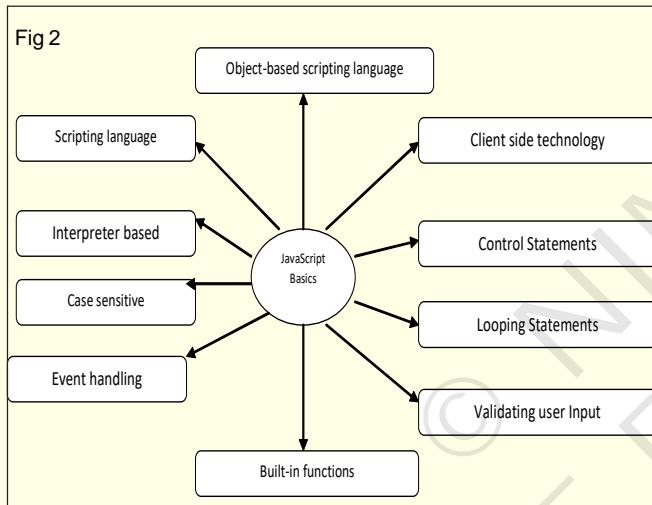
To start with, a text editor to write the code and a browser to display the web pages. A text editor of choice including Notepad++, Visual Studio Code, Sublime Text, Atom or any other text editor is comfortable with. And also, can use any web browser including Google Chrome, Firefox, Microsoft Edge, Internet Explorer etc.

Sample JavaScript program

```
<html>
<head>
<title>My First JavaScript code!!!</title>
<script type="text/javascript">
alert("Welcome to JavaScript Program!");
</script>
</head>
<body>
</body>
</html>
```

Features of JavaScript

JavaScript is a client side technology, it is mainly used for client side validation, but it has lot of features which are shown in Fig 2.



- JavaScript is a object-based scripting language.
- It gives the user more control over the browser.
- It Handles dates and time.
- It detects the user's browser and OS,
- It is light weighted.
- It is a scripting language and it is not java.
- It is interpreter based scripting language.
- It is case sensitive.
- It is object based language as it provides predefined objects.

- Every statement in JavaScript must be terminated with semicolon (;).
- Most of the JavaScript control statements syntax is same as syntax of control statements in C language.
- An important part of JavaScript is the ability to create new functions within scripts.

Advantages of JavaScript

- Executed on the client side: For example, user can validate any user input before sending a request to the server. This makes less load on the server.
- Relatively an easy language: This is quite easy to learn and the syntax that is close to English.
- Instant response to the visitors: Without any server interaction, don't have to wait for a page reload to get the desire result.
- Fast to the end user: As the script is executed on the user's computer, depending on task, the results are completed almost instantly.
- Interactivity increased: Creating interfaces that can react when the user hovers over them or activates them using the keyboard.
- Rich interfaces: Drag and drop components or slider may give a rich interface to site visitors.

Disadvantages of JavaScript

- Security issues: Any JavaScript snippets, while appended onto web pages on client side immediately can also be used for exploiting the user's system.
- Doesn't have any multiprocessor or multi threading capabilities.
- As no supports are available, JavaScript cannot be used for any networking applications.
- JavaScript does not allow us to read or write files.
- JavaScript render varies: JavaScript may be rendered by different layout engines differently. As a result, this causes inconsistency in terms of interface and functionality.

JavaScript Versions

JavaScript was invented by Brendan Eich in 1995, and became an ECMA standard in 1997. ECMA Script is the official name of the language.

From 2015 ECMA Script is named by year (ECMA Script 2015).

ECMA Script Editions

Ver	Official Name	Description
1	ECMA Script 1 (1997)	First Edition.
2	ECMA Script 2 (1998)	Editorial changes only.
3	ECMA Script 3 (1999)	Added Regular Expressions. Added try/catch.
4	ECMA Script 4	Never released.
5	ECMA Script 5 (2009)	Added “strict mode”. Added JSON support. Added String.trim(). Added Array.isArray(). Added Array Iteration Methods.
5.1	ECMA Script 5.1 (2011)	Editorial changes.
6	ECMA Script 2015	Added let and const. Added default parameter values. Added Array.find().Added Array.findIndex().
7	ECMA Script 2016	Added exponential operator (**). Added Array.prototype.includes.
8	ECMA Script 2017	Added string padding. Added new Object properties. Added Async functions.Added Shared Memory.
9	ECMA Script 2018	Added rest/spread properties. Added Asynchronous iteration. Added Promise.finally().Additions to Reg Exp.

Introduction to Web servers and External JavaScript files

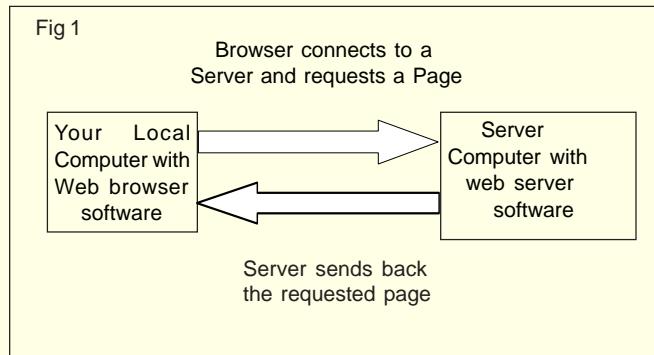
Objectives : At the end of this lesson you shall be able to

- explain Web servers and their features
- explain external JavaScript files.

What is Web Server and how it works?

Web servers are core for any web hosting. (Fig 1)

Web server is a program that uses **HTTP** to serve files that create web pages to users in response to their requests, which is sent by their computers HTTP connection. Any server that delivers an **XML document** to another device can be a web server. A better definition might be that a web server is an Internet server that responds to HTTP requests to deliver content and services. Always a web server is connected to the internet. Every web server that connects to the Internet will be having a **unique address** which contains a series of four numbers between 0 and 255. A period (.) separates these numbers. Also, web server enables the **hosting providers** to manage multiple domains(users) on a single server. A web host is a company that leases out space on a cluster of servers to empower people to serve their own content & webpages.



Different types of web servers

In open market there are different types of web servers available. Let's discuss about the most popular web servers. Apache, IIS, Nginx and Lite Speed are few of them.

Apache web server

One of the most popular web server in the world developed by the Apache Software Foundation. Apache is an open source software which supports almost all operating systems including Linux, Unix, Windows, FreeBSD, Mac OS X and more. About 60% of machines run on Apache Web Server. (Fig 2)

Fig 2



Customization of apache web server is easy as it contains a modular structure. It is also an open source which means that can add the own modules to the server when to require and make modifications that suit the requirements. It is more stable than any other web servers and is easier to solve administrative issues. It can be installed on multiple platforms successfully. Recent apache releases provide the feasibility of handling more requests when compare to its earlier versions.

IIS web server

IIS is a Microsoft product. This server has all the features just like apache. But it is not an open source and more over adding personal modules is not easy and modification becomes a little difficult job. (Fig 3)

Fig 3



Microsoft developed this product and they maintains, thus it works with all the windows operating system platforms. Also, they provides good customer support if it had any issues.

Nginx web server

Another free open source web server is Nginx, it includes IMAP/POP3 proxy server. Nginx is known for its high performance, stability, simple configuration and low resource usage. (Fig 4)

Fig 4



This web server doesn't use threads to handle requests rather a much more scalable event-driven architecture which uses small and predictable amounts of memory under load. It is getting popular in the recent times and it is hosting about 7.5% of all domains worldwide. Most of the web hosting companies are using this in recent times.

External JavaScript files: Writing JavaScript within HTML code sometimes creates confusion, and changing HTML files may also affect JavaScript files. So better to segregate HTML and JavaScript files so that, changes in one file does not affect other files.

The external JavaScript files should be written separately as follows:-

File `myjs.js` Contents:

```
function popup() {
    alert("Hello World");
}
```

Now we can import the file in HTML file as follows:-

Importing an external file is relatively painless. First, the file you are importing must be valid JavaScript, and only JavaScript. Second, the file must have the file extension ".js". Lastly, you must know the location of the file.

Let us assume we have a file "myjs.js" that contains a one line Hello World alert function. Also, let us assume that the file is the same directory as the HTML file we are going to code up. To import the file you would do the following in your HTML document.

JavaScript Code:

```
<html>
<head>
<script src="myjs.js">
</script>
</head>
<body>
<input type="button" onclick="popup()" value="Click Me!">
</body>
</html>
```

Now this HTML file imports myjs.js file and as a result it can access `popup()` function from HTML button element.

Using JavaScript Variable and data types

Objectives : At the end of this lesson you shall be able to

- explain variables in JavaScript
- explain various data types in JavaScript.

Variables

JavaScript variables are containers for storing data values.

In example 1, a, b, and c, are variables:

Example 1

```
var a = 12;
var b = 10;
var c = a + b;
```

From example 1, we can understand that

- **a** stores the value 12
- **b** stores the value 10
- **c** stores the value 22

In example 2, mark1, mark2, and total, are variables:

Example 2

```
var mark1 = 85;
var mark2 = 66;
var total = marks1 + mark2;
```

In programming, just like in algebra, we use variables **mark1** and **mark2** to hold values and use variables in expressions like **total = mark1 + mark2**. From the example above, and calculate the total to be 151.

JavaScript Identifiers

All JavaScript **variables** must be **identified** with **unique names**. These unique names are called **identifiers**. Identifiers can be short names like a and b or more descriptive names like mark1, mark2, total, age, sum, total volume.

The general rules for constructing names for variables are:

- Names can contain letters, digits, underscores, and dollar signs.
- Names must begin with a letter
- Names can also begin with \$ and _
- Names are case sensitive (a and A are different variables)
- Reserved words like JavaScript keywords cannot be used as names

Note: JavaScript identifiers are case-sensitive.

The Assignment Operator

In JavaScript, the equal sign (=) is an “assignment” operator, not an “equal to” operator.

```
x = x + 10;
```

It assigns the value of **x + 10** to **x**. It calculates the value of **x + 10** and puts the result into **x**. The value of **x** is incremented by 10.

JavaScript Data Types

JavaScript variables can hold numbers like 100 and text values like “Santhosh kumar”.

In programming, text values are called text strings. JavaScript can handle many types of data, but for now, just think of numbers and strings. **Strings** are written inside double or single quotes. **Numbers** are written without quotes. If you put a number in quotes, it will be treated as a text string.

Example 3

```
var pi = 3.14;
var person = "santhoshkumar";
var city = "coimbatore";
```

Declaring JavaScript Variables

Creating a variable in JavaScript is called **declaring** a variable. JavaScript variable is declared with the **var** keyword.

```
var traineeName;
```

After the declaration, the variable has no value. Technically it has the value of **undefined**. To **assign** a value to the variable, use the equal signs.

```
traineeName = "Santhosh Kumar";
```

You can also assign a value to the variable when you declare it.

```
var traineeName = "Santhosh Kumar";
```

In the example below, we create a variable called **traineeName** and assign the value “Santhosh Kumar” to it.

Then we “output” the value inside an HTML paragraph with **id="demo"**:

```
<p id ="demo"></p>
<script>
var traineeName = "santhoshkumar";
document.get ElementById ("demo").inner
HTML = traineeName;
</script>
```

Note: It is a good programming practice to declare all variables at the beginning of a script.

You can declare many variables in one statement. Start the statement with **var** and separate the variables by **comma**.

Example 4

```
var traineeName = "santhoshkumar", city =  
"coimbatore", total="151";
```

Undefined value

In computer programs, variables are often declared without a value. The value can be something that has to be calculated, or something that will be provided later, like user input.

A variable declared without a value will have the value **undefined**.

The variable **traineeName** will have the value **undefined** after the execution of this statement.

Example 5

```
var traineeName;
```

Re-Declaring JavaScript Variables

If you re-declare a JavaScript variable, it will not lose its value. The variable **traineeName** will still have the value "santhoshkumar" after the execution of these statements.

Example 6

```
var traineeName = "santhoshkumar";  
var traineeName;
```

JavaScript Arithmetic

Do the arithmetic with JavaScript variables, using operators like **=** and **+**

Example 7

```
var x = 8 + 2 + 5;
```

Now x has the value **15**.

You can also add strings, but strings will be concatenated:

Example 8

```
var x = "Dharani" + " " + "Shree"
```

Now x has the value **Dharani Shree**

The result of the following example gives **725**.

Example 9

```
var x = "7" + 2 + 5;
```

Note: If you put a number in quotes, the rest of the numbers will be treated as strings, and concatenated.

The result of the following example gives **75**.

Example 10

```
var x = 3 + 4 + "5";
```

Data types

In programming, data types is an important concept. To be able to operate on variables, it is important to know about the data type.

JavaScript variables can hold many **data types** like numbers, strings, objects and more.

Example 11

```
var side = 10; // Number  
var firstName = "Rithika"; // String  
var x = {firstName:"Harini", lastName:"Kumar"}; // Object
```

Without data types, a computer cannot safely solve this.

Example 12

```
var a = 10 + "Apple";
```

JavaScript will treat the example above as,

```
var a = "10" + "Apple";
```

The output is **10 Apple**

Note: When adding a number and a string, JavaScript will treat the number as a string.

JavaScript evaluates expressions from left to right. Different sequences can produce different results.

Example 13

```
var y = 20 + 5 + "Apple";
```

The result is **25Apple**

Example 14

```
var y = "Apple"+20 + 5 ;
```

The result is **Apple205**.

Note: In the first example, JavaScript treats 20 and 5 as numbers, until it reaches "Apple". In the second example, since the first operand is a string, all operands are treated as strings.

Dynamic data types

JavaScript has dynamic types. This means that the same variable can be used to hold different data types:

Example 15

```
var z; // Now z is undefined
```

```
z = 10; // Now z is a Number
```

```
z = "Sakthi"; // Now z is a String
```

JavaScript Strings

A string or a text string is a series of characters like "Harini Kumar". Strings are written with quotes. You can use single or double quotes.

Example 16

```
var bikeName = "Yamaha R15"; // Using double quotes  
var bikeName = ' Yamaha R15'; // Using single quotes
```

You can use quotes inside a string, as long as they don't match the quotes surrounding the string:

Example 17

```
var answer = "It's OK"; // Single quote inside  
                        double quotes  
  
var answer = 'Patel is called // Double quotes inside  
"Iron Man"';  
single quotes
```

JavaScript Numbers

JavaScript has only one type of numbers. Numbers can be written with or without decimals.

Example 18

```
var num1 = 87.0; // Written with decimals  
  
var num2 = 87; // Written without decimals
```

Extra large or extra small numbers can be written with scientific (exponential) notation:

Example 19

```
var exp1 = 232e5; // result is 23200000  
  
var z = 123e-5; // result is 0.00232
```

Example 20

```
var p = 3;  
  
var q = 3;  
  
var r = 5;  
  
(p == q) // Returns true  
(p == r) // Returns false
```

Note : Booleans are often used in conditional testing.

JavaScript Arrays

JavaScript arrays are written with square brackets. Array items are separated by commas. The following code declares (creates) an array called bikes, containing three items (bike names):

Example 21

```
var bikes = ["Yamaha", "TVS", "Royal Enfield"];
```

Note: Array indexes are zero-based, which means the first item is [0], second is [1], and so on.

JavaScript Objects

JavaScript objects are written with curly braces. Object properties are written as name:value pairs, separated by commas.

Example 22

```
var personName = {firstName:"Harini",lastName:  
                 "Kumar", age:13,height:  
                 "155 cms"};
```

The object (personName) in the example 22 above has 4 properties: firstName, lastName, age and height.

The typeof Operator

The JavaScript **typeof** operator is used to find the type of a JavaScript variable.

The **typeof** operator returns the type of a variable or an expression.

Example 23

typeof ""	// Returns "string"
typeof "Rithika"	// Returns "string"
typeof "Harini Kumar"	// Returns "string"
typeof 0	// Returns "number"
typeof 81	// Returns "number"
typeof 8.14	// Returns "number"
typeof(3+2)	// Returns "number"

Undefined

In JavaScript, a variable without a value, has the value **undefined**. The typeof is also **undefined**.

Example 24

```
var bike; // Value is undefined, type is  
           undefined
```

Note : Any variable can be emptied, by setting the value to undefined. The type will also be undefined.

Empty Values

An empty value has nothing to do with undefined. An empty string has both a legal value and a type.

Example 25

```
var bike = ""; // The value is "", the typeof  
               is "string"
```

Null

In JavaScript null is "nothing". It is supposed to be something that doesn't exist. In JavaScript, the data type of null is an object. You can empty an object by setting it to null.

Example 26

```
var personName = {firstName:"Harini",lastName:  
                 "Kumar", age:13,height:"155 cms"};  
  
personName = null; //Now value in null, but  
                   type is still an object
```

You can also empty an object by setting it to undefined:

Example 27

```
var personName = {firstName:"Harini", lastName:  
    "Kumar", age:13, height:"155 cms"};  
  
personName = undefined; // Now both value and  
type is undefined.
```

Difference Between Undefined and Null

Undefined and null are equal in value but different in type.

Example 28

```
typeof undefined // undefined  
typeof null // object  
null === undefined // false  
null == undefined // true
```

Primitive Data

A primitive data value is a single simple data value with no additional properties and methods. The **typeof** operator can return one of these primitive types.

- string
- number
- boolean
- undefined

Example 29

```
typeof "Rajesh" // Returns "string"  
typeof 1.44 // Returns "number"  
typeof true // Returns "boolean"  
typeof false // Returns "boolean"  
typeof a // if a has no value, it returns  
"undefined"
```

Complex Data

The **typeof** operator can return one of two complex types:

- function
- object

The type of operator returns object for both objects, arrays and null. It does not return object for functions.

Example 30

```
typeof {name, 'Karthik', age 27} // Returns "object"  
typeof [10, 20, 30, 40, 50] // Returns "object"  
// (not "array", see  
// note below)  
typeof null // Returns "object"  
typeof function sampleFunc() {} // Returns "function"
```

Note: The **typeof** operator returns "object" for arrays because in JavaScript arrays are objects.

Using JavaScript Constants and Operators

Objectives : At the end of this lesson you shall be able to

- explain constants in JavaScript
- explain operators in JavaScript.

Constants

Constants are a special kind of variable, store a value that never changes during the course of the program.

The syntax to create a Constant is.

```
const CONSTANT_NAME:DataType = value;
```

In the above syntax, "**const**" is the special keyword, reserved to define a constant. As you can see, this syntax looks a lot like a variable declaration but with the **var** keyword replaced with "const". Most programmers use all caps for the name of the constants to differentiate them from variables.

Example 1

```
const FRIEND = 'Shanthy';  
const BROTHER_AGE = 46;
```

Note :The keyword const is a little misleading.
It does NOT define a constant value. It defines a constant reference to a value.Because of this, we cannot change constant primitive values, but we can change the properties of constant objects.

Primitive Values

If we assign a primitive value to a constant, we cannot change the primitive value.

Example 2

```
const PI = 3.141592653589793;  
  
PI = 3.14; // This will give an error  
  
PI = PI + 10; // This will also give an error
```

Constant Objects can Change

Change the properties of a constant object.

Example 3

```
// You can create a const object:  
const bike = {type: "Yamaha", model: "R15", color:  
    "blue"};  
  
// You can change a property:  
bike.color = "grey";
```

```
// You can add a property:  
bike.owner = "Sree";
```

But you can NOT reassign a constant object.

Example 4

```
const bike = {type:"Yamaha", model:"R15", color:"blue"};  
bike = {type:"Tvs", model:"Star city", color:"black"};  
// ERROR
```

Constant Arrays can Change

You can change the elements of a constant array.

Example 5

// You can create a constant array:

```
constant bikes = ["TVS", "Yamaha", "Royal  
Enfield"];
```

// You can change an element;

```
bikes[0] = "suzuki";
```

```
bikes.push ("Bajaj"); //you can add an element
```

But you can NOT reassign a constant array:

Example 6

```
const bikes = {"TVS", "Yamaha", "Royal Enfield"};  
bikes = ["TVS", "Yamaha", Bajaj"]; //ERROR
```

Operators

There are eight types of operators in JavaScript. These are

- Additive Operators
- Multiplicative Operators
- Bitwise operator
- Equality operator
- Relational Operator
- Unary Operators
- Ternary Operator
- Assignment Operators

Additive Operators: The term additive operators include both addition (+) and subtraction(-) as subtraction is also addition with a negative number.

Example 7

```
32+67; // this is 99  
d+e; // Adds d with e  
3-7; // return -4
```

Sometimes JavaScript addition can results in unexpected results.

Example 8

```
var num1="Runs";  
var num2=784;
```

```
var res=num1+num2; // result is Runs784 as java concats  
two strings.
```

Multiplicative Operators: Just like Additive operators, multiplication(*), division(/) and modulo(%) are multiplicative Operators. Modulo operator return remainder of division.

Example 9

```
javascript:alert(4%3); // returns 1
```

Additive and Multiplicative operator together can be called Arithmetic Operator.

Bitwise operator:

JavaScript Uses 32 bits Bitwise Operands. JavaScript stores numbers as 64 bits floating point numbers, but all bitwise operations are performed on 32 bits binary numbers. Before a bitwise operation is performed, JavaScript converts numbers to 32 bits signed integers. After the bitwise operation is performed, the result is converted back to 64 bits JavaScript numbers.

Operator	Name	Description
&	AND	Sets each bit to 1 if both bits are 1
	OR	Sets each bit to 1 if one of two bits is 1
^	XOR	Sets each bit to 1 if only one of two bits is 1
~	NOT	Inverts all the bits
<<	Zero fill left shift	Shifts left by pushing zeros in from the right and let the leftmost bits fall off
>>	Signed right shift	Shifts right by pushing copies of the leftmost bit in from the left, and let the rightmost bits fall off
>>>	Zero fill right shift	Shifts right by pushing zeros in from the left, and let the rightmost bits fall off

Example10

Operation	Result	Same as	Result
5 & 1	1	0101 & 0001	0001
5 1	5	0101 0001	0101
~ 5	10	~0101	1010
5 << 1	10	0101 << 1	1010
5 ^ 1	4	0101 ^ 0001	0100
5 >> 1	2	0101 >> 1	0010
5 >>> 1	2	0101 >>> 1	0010

Note: The examples above uses 4 bits unsigned binary numbers. Because of this `~ 5` returns 10.

Since JavaScript uses 32 bits signed integers, it will not return 10. It will return -6.

`0000000000000000000000000000000101 (5)`

`111111111111111111111111111111010 (~5 = -6)`

A signed integer uses the leftmost bit as the minus sign.

Equality operator

Equality operator are used to test whether two expressions are the same.

For example "42" and 42 are equal with `==` but not equal with `===` as it not only checks for value but also check for type.

Relational Operator

Relational operator checks whether one value is greater or less than other value.

```
if(5<2) {
    // do something
}
```

Operator	Function
<code>></code>	Greater than
<code><</code>	Less than
<code>>=</code>	Greater than or equal to
<code><=</code>	Less than or equal to
<code>in</code>	Tests whether a value is found in an expression
<code>instanceof</code>	Tests whether an expression is an instance of an object

The less than operator checks whether the first value is less than second value and if valid, it returns false. In the above example, 5 is not less than 2, so it is not true and code inside the if block will not execute.

The other three operator are in the same way do checking for greater then (`>`), Greater than or equal to (`>=`), Less than or equal to (`<=`).

`in` operator checks whether a given index is contained within an object. For example:

```
var MyObj= {star:"Algol", constellation: "Perseus"};
if("star" in MyObj) {
    // do something
}
```

As `star` is a index the code will work. but `in` operator do not work on numeric types as it works for numbers only.

Instanceof Operator checks whether an object instance or object variable of is an instance of a particular object.

Example 11

```
var mydate=new Date();
if(mydate instanceof Date) {
    //do something
}
```

Here `mydate` is an instance of built-in `Date` object. So the code will be executed within the if block.

Unary Operator

`delete`, `void`, `typeof`, `++`, `--`, `+ -`, `~`, `!` are unary operators in Javascript.

Example 12

```
a = -10;
p=++a;
q=a++;
s+=p;
```

There are pre and post increment and decrement operator.

`p=++a;` is equivalent to

```
a=a+1;
p=a;
```

and `q=a++;` is equivalent to

```
q=a;
a=a+1;
```

The delete Operator

The `delete` operator can be used to delete properties from objects.

Example 13

```
var person = {firstName:"John", lastName:"Doe", age:50,
eyeColor:"blue"};
delete person.age;
```

The `delete` operator is designed to be used on object properties. It has no effect on variables or functions.

The `delete` operator should not be used on predefined Java Script object properties. It can crash your application.

The Unary + Operator

The unary `+` operator can be used to convert a variable to a number.

Example 14

```
var y = "5";    // y is a string
var x = + y;   // x is a number
```

If the variable cannot be converted, it will still become a number, but with the value `Nan` (Not a number):

Example 15

```
var y = "John"; // y is a string
var x = + y;   // x is a number (NaN)
```

In the same way Unary - Operator also operates.

Ternary or Conditional Operator: It can be used as compact if else.

Example 16

a = (b>5 ? 4:7); means

if(b>5)

a=4;

else

a=7;

Assignment Operator:

Assignment Operator is used to assign values into a variable. Apart from = there are compound assignment operators as follows-

*=	/=	%=
+=	-=	<<=
>>=	>>>=	&=
^=	=	

a = q; means value of q is assigned into a variable deleting the previous value of a.

Now a* = 3; is equivalent to a = a*3; and like that all other compound assignment operator behaves.

Control statements, Loops and Popup boxes in JavaScript

Objectives : At the end of this lesson you shall be able to

- explain control statements
- discuss about various Loops
- explain the uses of Popup boxes.

Control Statements: When we write code for a particular program, we sometimes takes various decisions for executing different action. These can be done through conditional/control statements.

In JavaScript we have the following conditional statements:

Use **if** to specify a block of code to be executed, if a specified condition is true

Use **else** to specify a block of code to be executed, if the same condition is false

Use **else if** to specify a new condition to test, if the first condition is false

Use **switch** to specify many alternative blocks of code to be executed.

The if Statement

Use the if statement to specify a block of JavaScript code to be executed if a condition is true.

Syntax

```
If (condition) {
    block of code to be executed if the condition is true
}
```

Example 1

Make a "Good day" greeting if the time is less than 18:00:

```
if (time < 18) {
    greeting = "Good day";
}
```

The **result** of greeting will be:

Good day

The else Statement

Use the else statement to specify a block of code to be executed if the condition is false.

```
if (condition) {
    block of code to be executed if the condition is true
} else {
    block of code to be executed if the condition is false
}
```

Example 2

If the time is less than 18:00, create a "Good day" greeting, otherwise "Good evening":

```
if (time < 18) {
    greeting = "Good day";
} else {
    greeting = "Good evening";
}
```

The **result** of greeting will be:

Good day

The else if Statement

Use the else if statement to specify a new condition if the first condition is false.

Syntax

```
if (condition1) {
    block of code to be executed if condition1 is true
} else if (condition2) {
    block of code to be executed if the condition1 is false
        and condition2 is true
} else {
    block of code to be executed if the condition1 is false
        and condition2 is false
}
```

Example 3

If time is less than 10:00, create a "Good morning" greeting, if not, but time is less than 18:00, create a "Good day" greeting, otherwise a "Good evening":

```
if (time < 10) {
    greeting = "Good morning";
} else if (time < 18) {
    greeting = "Good day";
} else {
    greeting = "Good evening";
}
```

The **result** of x will be:

Good day

The JavaScript Switch Statement

Use the switch statement to select one of many blocks of code to be executed.

Syntax

```
switch(expression) {  
    case n1:  
        code block  
        break;  
    case n2:  
        code block  
        break;  
    default:  
        default code block  
}
```

This is how it works:

- The switch expression is evaluated once.
- The value of the expression is compared with the values of each case.
- If there is a match, the associated block of code is executed.

Example 4

Use today's weekday number to calculate weekday name:
(Sunday=0, Monday=1, Tuesday=2, ...)

```
switch (new Date().getDay()) {  
    case 0:  
        day = "Sunday";  
        break;  
    case 1:  
        day = "Monday";  
        break;  
    case 2:  
        day = "Tuesday";  
        break;  
    case 3:  
        day = "Wednesday";  
        break;  
    case 4:  
        day = "Thursday";  
        break;  
    case 5:  
        day = "Friday";  
        break;  
    case 6:  
        day = "Saturday";  
        break;  
}
```

The **result** of day will be:

Tuesday

The break Keyword

When the JavaScript code interpreter reaches a break keyword, it breaks out of the switch block.

This will stop the execution of more execution of code and/or case testing inside the block.

The default Keyword

The default keyword specifies the code to run if there is no case match:

Example 5

If today is neither Saturday nor Sunday, write a default message:

```
switch (new Date().getDay()) {  
    case 6:  
        text = "Today is Saturday";  
        break;  
    case 0:  
        text = "Today is Sunday";  
        break;  
    default:  
        text = "Looking forward to the Weekend";  
}
```

The **result** of text will be:

Looking forward to the Weekend

Common Code and Fall-Through

Sometimes, in a switch block, you will want different cases to use the same code, or fall-through to a common default.

Note from the next example, that cases can share the same code block and that the default case does not have to be the last case in a switch block:

Example 6

```
switch (new Date().getDay()) {  
    case 1:  
    case 2:  
    case 3:  
    default:  
        text = "Weekend is coming";  
        break;  
    case 4:  
    case 5:  
        text = "Weekend is soon";  
        break;  
    case 0:
```

```

case 6:
    text = "Now in Weekend";
}

```

JavaScript Loops

Loops are handy, if you want to run the same code over and over again, each time with a different value.

Often this is the case when working with arrays:

Instead of writing:

```

text += train[0] + "<br>";
text += train [1] + "<br>";
text += train [2] + "<br>";
text += train [3] + "<br>";
text += train [4] + "<br>";
text += train [5] + "<br>";

```

You can write:

```

for (i = 0; i < train.length; i++) {
    text += train [i] + "<br>";
}

```

Different Kinds of Loops

JavaScript supports different kinds of loops:

- for - loops through a block of code a number of times
- for/in - loops through the properties of an object
- while - loops through a block of code while a specified condition is true
- do/while - also loops through a block of code while a specified condition is true

The For Loop

The for loop is often the tool you will use when you want to create a loop.

The for loop has the following syntax:

```

for (statement 1; statement 2; statement 3) {
    code block to be executed
}

```

Statement 1 is executed before the loop (the code block) starts. It is called Initialisation Part

Statement 2 defines the condition for running the loop (the code block). It is called condition part.

Statement 3 is executed each time after the loop (the code block) has been executed. It is called increment/decrement part.

Example 7

```

for (i = 0; i < 5; i++) {
    text += "The number is " + i + "<br>";
}

```

From the example above, you can read:

Statement 1 sets a variable before the loop starts (var i = 0).

Statement 2 defines the condition for the loop to run (i must be less than 5).

Statement 3 increases a value (i++) each time the code block in the loop has been executed.

Initialisation Part

Normally you will use statement 1 to initiate the variable used in the loop (var i = 0).

This is not always the case, JavaScript doesn't care. Statement 1 is optional.

You can initiate many values in statement 1 (separated by comma):

Example 8

```

for (i = 0, len = train.length, text = ""; i < len; i++) {
    text += train [i] + "<br>";
}

```

And you can omit statement 1 (like when your values are set before the loop starts):

Example 9

```

var i = 2;
var len = train.length;
var text = "";
for (; i < len; i++) {
    text += train [i] + "<br>";
}

```

Condition Part

Often statement 2 is used to evaluate the condition of the initial variable.

This is not always the case, JavaScript doesn't care. Statement 2 is also optional.

If statement 2 returns true, the loop will start over again, if it returns false, the loop will end.

If you omit statement 2, you must provide a break inside the loop. Otherwise the loop will never end. This will crash your browser. Read about breaks in a later chapter of this tutorial.

Increment/Decrement Part

Often statement 3 increases the initial variable.

This is not always the case, JavaScript doesn't care, and statement 3 is optional.

Statement 3 can do anything like negative increment (i--), or larger increment (i = i + 15), or anything else.

Statement 3 can also be omitted (like when you increment your values inside the loop):

Example 10

```
var i = 0;
len = train.length;
for (; i < len; ) {
    text += train [i] + "<br>";
    i++;
}
```

For/In Loop

The JavaScript for/in statement loops through the properties of an object:

```
var person = {fname:"Raja", lname:"Sen", age:35};
var text = "";
var x;
for (x in person) {
    text += person[x];
}
```

While Loop

The while loop loops through a block of code as long as a specified condition is true.

Syntax

```
while (condition) {
    code block to be executed
}
```

In the following example, the code in the loop will run, over and over again, as long as a variable (i) is less than 10:

Example 11

```
while (i < 10) {
    text += "The number is " + i;
    i++;
}
```

If you forget to increase the variable used in the condition, the loop will never end. This will crash your browser.

The Do/While Loop

The do/while loop is a variant of the while loop. This loop will execute the code block once, before checking if the condition is true, then it will repeat the loop as long as the condition is true.

Syntax

```
do {
    code block to be executed
}

while (condition);
```

The example below uses a do/while loop. The loop will always be executed at least once, even if the condition is false, because the code block is executed before the condition is tested:

Example 12

```
do {
    text += "The number is " + i;
    i++;
}
while (i < 10);
```

Do not forget to increase the variable used in the condition, otherwise the loop will never end!

Comparing For and While

If you have read the previous chapter, about the for loop, you will discover that a while loop is much the same as a for loop, with statement 1 and statement 3 omitted.

The loop in this example uses a for loop to collect the car names from the train array:

Example 13

```
train = ["Duronto", "Satabdi", "Garib Rath", "Rajdhani"];
var i = 0;
var text = "";
for (;train[i]); {
    text += train[i] + "<br>";
    i++;
}
```

The loop in this example uses a while loop to collect the car names from the train array:

```
train = ["Duronto", "Satabdi", "Garib Rath", "Rajdhani"];
var i = 0;
var text = "";
while (train[i]) {
    text += train[i] + "<br>";
    i++;
}
```

The Break Statement in Loop

Break statement is used to terminate a loop before its completion. It saves machine time for not iterating a loop uselessly.

For example: In linear search, if we find the item then we can break the loop as no point of running it unnecessary.

Example 14

```
for(i=0;i<l;i++) {
    if(arr[i]==item) {
        alert("Found at :" + i);
        fl=1;
        break;
    }
}
if(fl==0) alert("Not Found");
```

Here, if the item is found, loop breaks and CPU time is saved.

Popup Boxes

JavaScript has three kind of popup boxes. They are

- 1 Alert box
- 2 Confirm box and
- 3 Prompt box.

Alert Box

An alert box is often used if you want to make sure information comes through to the user. When an alert box pops up, the user will have to click "OK" to proceed.

Syntax

```
window.alert("sometext");
```

Note: The window.alert() method can be written without the window prefix.

Example 15

```
alert ("Welcome to Java Script Coding!");
```

The result is shown in Fig 1.



Confirm Box

A confirm box is often used to verify or accept something. When a confirm box pops up, the user will have to click either "OK" or "Cancel" to proceed. If the user clicks "OK", the box returns **true**. If the user clicks "Cancel", the box returns **false**.

Syntax

```
window.confirm("sometext");
```

Note: The window.confirm() method can be written without the window prefix.

Example 16

```
if(confirm("Click a button!"))
{
    txt = " You clicked OK!";
}
else
{
    txt = "You clicked Cancel!";
}
```

The result is is shown in Fig 2.

Fig 2



Note: When click on OK button it displays the message "You clicked OK!" and when click on Cancel button it displays the message "You clicked Cancel!"

Prompt Box

A prompt box is often used if the user to input a value before entering a page. When a prompt box pops up, the user will have to click either "OK" or "Cancel" to proceed after entering an input value. If the user clicks "OK" the box returns the input value. If the user clicks "Cancel" the box returns null.

Syntax

```
window.prompt("sometext", "default text");
```

Note: The window.prompt() method can be written without the window prefix.

Example 17

```
var tname = prompt("Please Enter your Name", "Lakshmi");
```

```
if (tname == null || tname == "")
{
    txt = "User cancelled the prompt.";
}
else
{
    txt = "Hello " + tname + "! Congratulations!!!!";
}
```

The result is is shown in Fig 3.

Fig 3



Note: If click on OK button it displays the message "Hello Lakshmi! Congratulations!!!!" If cancelled the name 'Lakshmi' as shown in Fig 4 it gives the message "User cancelled the Prompt". Also when click the Cancel button even when if the box has text 'Lakshmi' it gives the message "User cancelled the Prompt".

Fig 4



Line Breaks

To display line breaks inside a popup box, use a backslash followed by the character n.

Example 18

```
alert("Hello\nWelcome!");
```

The result is shown in Fig 5.

Fig 5



Arrays in JavaScript

Objectives : At the end of this lesson you shall be able to

- define Array
- explain concepts of Array
- describe array methods
- know sorting of Array.

What is an Array?

An array is a special variable, which can hold more than one value at a time.

If you have a list of items (a list of train names, for example), storing the trains in single variables could look like this.

```
var train1 = "Jan Satabdi";
var train1 = "Garib Rath";
var train1 = "Duronto";
```

However, what if you want to loop through the trains and find a specific one? And what if you had not 3 trains, but 300?

The solution is an array!

JavaScript Arrays

JavaScript arrays are used to store multiple values in a single variable.

Creating an Array

Using an array literal is the easiest way to create a JavaScript Array.

Syntax:

```
var array-name = [item1, item2, ...];
```

Example 1

```
var trains = ["Duronto", "Jan Satabdi", "RAJDHANI"];
```

Using the JavaScript Keyword new

The following example also creates an Array and assigns values to it:

Example 2

```
var trains = new Array("Duronto", "Jan Satabdi",
"RAJDHANI");
```

The two examples above do exactly the same. There is no need to use new Array().

For simplicity, readability and execution speed, use the first one (the array literal method).

Access the Elements of an Array

You refer to an array element by referring to the index number.

This statement access the value of the first element in myTrains:

```
var name = trains[0];
```

This statement modifies the first element in trains:

```
trains[0] = "Jan Satabdi";
```

[0] is the first element in an array. [1] is the second. Array indexes start with 0.

Displaying Arrays: We will use a script to display arrays inside a `<p>` element with id="demo":

Example 3

```
<p id="demo"></p>
<script>
var trains = ["Duronto", "Jan Satabdi", "RAJDHANI"];
document.getElementById("demo").innerHTML = trains;
</script>
```

The first line (in the script) creates an **array** named **trains**.

The second line "finds" the element with id="demo", and "displays" the array in the "innerHTML" of it.

Spaces and line breaks are not important. A declaration can span multiple lines.

Example 4

```
var trains = [
"Duronto",
"Jan Satabdi",
"RAJDHANI"
];
```

Don't put a comma after the last element (like "RAJDHANI"). It is inconsistent across browsers.

An array can hold many values under a single name and you can access the values by referring to an index number.

You can have different objects in one array

JavaScript variables can be objects. Arrays are special kinds of objects.

Because of this, you can have variables of different types in the same Array.

You can have objects in an Array. You can have functions in an Array. You can have arrays in an Array:

```
my Array[0] = Date.now;
my Array [1] = my Function;
my Array [2] = myTrains;
```

Arrays are Objects

Arrays are a special type of objects. The `typeof` operator in JavaScript returns "object" for arrays.

But, JavaScript arrays are best described as arrays.

Arrays use numbers to access its "elements". In this example, `person[0]` returns Raja:

Object

```
var person = ["Raja", "Sen", 46];
```

Objects use names to access its "members". In this example, `person.firstName` returns Raja:

Length Property

The length property of an array returns the length of an array (the number of array elements).

Example 6

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];
```

```
fruits.length; // the length of fruits is 4
```

The length property is always one more than the highest array index.

Adding Array Elements

The easiest way to add a new element to an array is to use the length property:

Example 7

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];
fruits[fruits.length] = "Lemon"; // adds a new element (Lemon) to fruits
```

Adding elements with high indexes can create undefined "holes" in an array:

Example 8

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];
fruits[10] = "Lemon"; // adds a new element (Lemon) to fruits
```

Looping Array Elements

The best way to loop through an array is using a standard for loop:

Example 9

```
var index;
var fruits = ["Banana", "Orange", "Apple", "Mango"];
for (index = 0; index < fruits.length; index++) {
    text += fruits[index];
}
```

Associative Arrays? No Way!

Many programming languages support arrays with named indexes.

Arrays with named indexes are called associative arrays (or hashes).

JavaScript does not support arrays with named indexes.

Wrong:

```
var person = new Array()
person ["firstName"] = "Raja";
person ["lastName"] = "Sen";
person ["age"] = 46;
```

The example above looks like it works. But it does not.

If you try it, `person["firstName"]` will return Raja, but `person[0]` will return undefined, and `person.length` will return 0.

If you want to create an associative array, create an object instead.

When to Use Arrays? When to use Objects?

JavaScript does not support associative arrays.

You should use objects when you want the element names to be strings.

You should use arrays when you want the element names to be sequential numbers.

Avoid new Array()

There is no need to use the JavaScript's built-in array constructor `new Array()`.

Use [] instead.

These two different statements both create a new empty array named points.

```
var points = new Array(); // Bad
var points = []; // Good
```

These two different statements both create a new array containing 6 numbers.

```
var points = new Array(40, 100, 1, 5, 25, 10) // Bad
var points = [40, 100, 1, 5, 25, 10]; // Good
```

The `new` keyword complicates your code and produces nasty side effects.

```
var points = new Array(40, 100); // Creates an array with two elements (40 and 100)
```

What if I remove one of the elements?

```
var points = new Array(40); // Creates an array with 40 undefined elements !!!!!
```

How to Recognize an Array?

A common question is: How do I know if a variable is an array?

The problem is that the JavaScript operator `typeof` returns "object":

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];
```

```
typeof fruits; // typeof returns object
```

The type of operator returns object because a JavaScript array is an object.

To solve this problem you can create your own isArray() function:

```
function isArray(myArray) {  
    return myArray.constructor.toString().indexOf("Array") > 1;  
}
```

The function above always return true if the argument is an array.

Or more precisely: it returns true if the object proto type of the argument is "[object array]".

JavaScript Array Methods

Converting Arrays to Strings

toString() method

The JavaScript method **toString()** converts an array to a string of (comma separated) array values.

Example 10

```
var trade = ["COPA", "IT", "ICTSM", "CHNM", "Fitter"];  
document.getElementById("demo").innerHTML =  
trade.toString();
```

Result

COPA,IT,ICTSM,CHNM,Fitter

join() method

The **join()** method also joins all array elements into a string. It behaves just like **toString()**, but in addition you can specify the separator.

Example 11

```
var trade = ["COPA", "IT", "ICTSM", "CHNM", "Fitter"];  
document.getElementById("demo").innerHTML =  
trade.join("-");
```

Result

COPA - IT - ICTSM - CHNM - Fitter

Popping and Pushing

When you work with arrays, it is easy to remove elements and add new elements.

Popping

The **pop()** method removes the last element from an array.

Example 12

```
var trade = ["COPA", "IT", "ICTSM", "CHNM", "Fitter"];  
trade.pop(); // Removes the last element ("Fitter") from  
trade.
```

Result

COPA,IT,ICTSM,CHNM

The **pop()** method returns the value that was "popped out".

Example 13

```
var trade = ["COPA", "IT", "ICTSM", "CHNM"];  
var x = trade.pop(); // the value of x is  
"CHNM".
```

Pushing

The **push()** method adds a new element to an array (at the end).

Example 14

```
var trade = ["COPA", "IT", "ICTSM", "CHNM"];  
trade.push("DTPO"); // Adds a new element ("DTPO")  
to trade.
```

Result

COPA,IT,ICTSM,CHNM,DTPO

The **push()** method returns the new array length.

Example 15

```
var trade = ["COPA", "IT", "ICTSM", "CHNM"];  
var x = trade.push("DTPO"); // the value of x is 5
```

Shifting Elements

Shifting is equivalent to popping, working on the first element instead of the last. The **shift()** method removes the first array element and "shifts" all other elements to a lower index.

Example 16

```
var trade = ["COPA", "IT", "ICTSM", "CHNM"];  
trade.shift(); // Removes the first element "COPA" from  
trade.
```

Result

IT,ICTSM,CHNM

The **shift()** method returns the string that was "shifted out":

Example 17

```
var trade = ["COPA", "IT", "ICTSM", "CHNM"];  
trade.shift(); // Returns "COPA"
```

The **unshift()** method adds a new element to an array (at the beginning), and "unshifts" older elements.

Example 18

```
var trade = ["COPA", "IT", "ICTSM", "CHNM"];  
fruits.unshift("ElecMech"); // Adds a new element  
"ElecMech" to trade
```

Result

ElecMech,COPA,IT,ICTSM,CHNM

The **unshift()** method returns the new array length.

Example 19

```
var trade = ["COPA", "IT", "ICTSM", "CHNM"];  
trade.unshift("ElecMech"); // Returns 5
```

Changing Elements

Array elements are accessed using their **index number**: Array **indexes** start with 0. [0] is the first array element, [1] is the second, [2] is the third ...

Example 20

```
var trade = ["COPA", "IT", "ICTSM", "CHNM"];
trade[2] = "DTPO"; // Changes the third element of
// trade to "DTPO"
```

Result

COPA,IT,DTPO,CHNM

The length property provides an easy way to append a new element to an array:

Example 21

```
var trade = ["COPA", "IT", "DTPO", "CHNM"];
trade[trade.length] = "ICTSM"; // Appends "ICTSM"
// to fruits
```

Result

COPA,IT,DTPO,CHNM,ICTSM

Deleting Elements

Since JavaScript arrays are objects, elements can be deleted by using the JavaScript operator **delete**.

Example 22

```
var trade = ["COPA", "IT", "DTPO", "CHNM"];
delete trade[0]; // Changes the first element in
// trade to undefined
```

Note :Using delete may leave undefined holes in the array. Use pop() or shift() instead

Splicing an Array

The **splice()** method can be used to add new items to an array:

Example 23

```
var trade = ["COPA", "IT", "DTPO", "CHNM"];
trade.splice(2, 0, "Turner", "Machinist");
```

Note: The first parameter (2) defines the position where new elements should be added (spliced in).The second parameter (0) defines how many elements should be removed.The rest of the parameters ("Turner", "Machinist")define the new elements to be added.

Result

COPA,IT,Turner,Machinist,DTPO,CHNM

Using splice() to Remove Elements

With clever parameter setting, you can use **splice()** to remove elements without leaving "holes" in the array.

Example 24

```
var trade = ["COPA", "IT", "DTPO", "CHNM"];
trade.splice(0, 1); // Removes the first element of
// trade
```

Result

IT,DTPO,CHNM

Note :The first parameter (0) defines the position where new elements should be added (spliced in). The second parameter (1) defines how many elements should be removed.The rest of the parameters are omitted. No new elements will be added.

Merging or Concatenating Arrays

The **concat()** method creates a new array by merging existing arrays.

Example 25 (Merging Two Arrays)

```
var names1 = ["Devi", "Deepa"];
var names2 = ["Poorna", "Saranya", "Shalini"];
var myTrainee = names1.concat(names2);
//Concatenates (joins) names1 and names2.
```

Result

Note: The concat() method does not change the existing arrays. It always returns a new array.

The concat() method can take any number of array arguments.

Example 26 (Merging Three Arrays)

```
var arr1 = ["Priya", "Mythili"];
var arr2 = ["Sangeetha", "Nancy", "Sahana"];
var arr3 = ["Ramya", "Kavi"];
var myTrainee = arr1.concat(arr2, arr3);
// Concatenates arr1 with arr2 and arr3
```

The concat() method can also take values as arguments.

Example 27 (Merging an Array with Values)

```
var arr1 = ["Priya", "Mythili"];
var myTrainee = arr1.concat(["Ramya", "Kavi"]);
```

Slicing an Array

The **slice()** method slices out a piece of an array into a new array. This example slices out a part of an array starting from array element 2 ("DTPO"). The slice() method creates a new array. It does not remove any elements from the source array.

Example 28

```
var trade = ["COPA", "IT", "DTPO", "CHNM"];
var trade1 = trade.slice(2);
```

The slice() method can take two arguments like slice(1, 3). The method then selects elements from the start argument, and up to (but not including) the end argument.

Example 29

```
var trade = ["COPA", "IT", "DTPO", "CHNM"];
var trade1 = trade.slice(1,3);
```

If the end argument is omitted, like in the first examples, the slice() method slices out the rest of the array.

Example 30

```
var trade = ["COPA", "IT", "DTPO", "CHNM"];
var trade1 = trade.slice(2);
```

Automatic `toString()`

JavaScript automatically converts an array to a comma separated string when a primitive value is expected. This is always the case when you try to output an array.

These two examples will produce the same result:

Example 31

```
var trade = ["COPA", "IT", "DTPO", "CHNM"];
document.getElementById("demo").innerHTML =
trade.toString();
```

Concepts of Animation and Multimedia files in JavaScript

Objectives : At the end of this lesson you shall be able to

- know animation settings in JavaScript
- explain multimedia in JavaScript.

Animation

Styling the Elements

To make an animation possible, the animated element must be animated relative to a "parent container".

The container element should be created with style = "position: relative".

The animation element should be created with style = "position: absolute".

Example

```
<!Doctype html>
<html>
<style>
#myContainer {
  width: 400px;
  height: 400px;
  position: relative;
  background: pink;
}
#myAnimation {
  width: 50px;
```

Example 32

```
var trade = ["COPA", "IT", "DTPO", "CHNM"];
document.getElementById("demo").innerHTML = trade;
```

Note: All JavaScript objects have a `toString()` method.

Sorting an Array

The `sort()` method sorts an array alphabetically.

Example 33

```
var trade = ["COPA", "IT", "ICTSM", "CHNM"];
trade.sort(); // Sorts the elements of trade
```

Result

CHNM,COPA,ICTSM,IT

Reversing an Array

The `reverse()` method reverses the elements in an array. You can use it to sort an array in descending order.

Example 34

```
var trade = ["COPA", "IT", "ICTSM", "CHNM"];
trade.sort(); // Sorts the elements of trade
trade.reverse(); // Reverse the order of the elements
```

```
height: 50px;
position: absolute;
background: green;
}
</style>
<body>
<h1>My First JavaScript Animation</h1>
<div id="myContainer">
<div id="myAnimation"></div>
</div>
</body>
</html>
?
```

The Animation Code

JavaScript animations are done by programming gradual changes in an element's style. The changes are called by a timer. When the timer interval is small, the animation looks continuous. The basic code is:

Example

```
var id = setInterval(frame, 5);
```

```

function frame() {
if /* test for finished */ {
clearInterval(id);
} else {
/* code to change the element style */
}
}

Create the Animation Using JavaScript

Example

<style>
#myContainer{
width: 400px;
height: 400px;
position: relative;
background: pink;
}
#myAnimation {
width: 50px;
height: 50px;
position: absolute;
background-color: green;
}
</style>
<body>
<p>
<button onclick="myMove()">Click Me</button>
</p>
<div id="myContainer">
<div id="myAnimation"></div>
</div>
<script>
function myMove() {
var elem=document.getElementById("myAnimation");
var pos = 0;
var id=setInterval(frame, 10);
function frame() {
if (pos == 350) {
clearInterval(id);
} else {
pos++;
elem.style.top = pos + 'px';
}
}
}

```

```

elem.style.left = pos + 'px';
}
}
}
</script>
</body>
</html>

```

Multimedia files

What is Multimedia?

Multimedia comes in many different formats. It can be almost anything you can hear or see. Web pages often contain multimedia elements of different types and formats.

Examples: Images, music, sound, videos, records, films, animations and more.

Multimedia Formats

Multimedia elements (like audio or video) are stored in media files. The most common way to discover the type of a file, is to look at the file extension. Multimedia files have formats and different extensions like: .swf, .wav, .mp3, .mp4, .mpg, .wmv, and .avi.

Playing Videos in HTML

To show a video in HTML, use the <video> element:

Example

```

<video width="320" height="240" controls>
<source src="movie.mp4" type="video/mp4">
<source src="movie.ogg" type="video/ogg">
Your browser does not support the video tag.
</video>

```

How it Works

The controls attribute adds video controls, like play, pause, and volume. It is a good idea to always include width and height attributes. If height and width are not set, the page might flicker while the video loads. The <source> element allows you to specify alternative video files which the browser may choose from. The browser will use the first recognized format. The text between the <video> and </video> tags will only be displayed in browsers that do not support the <video> element.

HTML <video> Autoplay

To start a video automatically use the autoplay attribute:

Example

```

<video width="320" height="240" autoplay>
<source src="movie.mp4" type="video/mp4">
<source src="movie.ogg" type="video/ogg">
Your browser does not support the video tag.
</video>

```

Note: Autoplay attribute does not work in mobile devices like iPad and iPhone

HTML Video - Media Types

File Format	Media Type
MP4	video/mp4
WebM	video/webm
Ogg	video/ogg

HTML Video - Methods, Properties, and Events

HTML5 defines DOM methods, properties, and events for the <video> element. This allows you to load, play, and pause videos, as well as setting duration and volume. There are also DOM events that can notify you when a video begins to play, is paused, etc.

HTML5 Video Tags

Tag	Description
<video>	Defines a video or movie
<source>	Defines multiple media resources for media elements, such as <video> and <audio>
<track>	Defines text tracks in media players

Audio on the Web

The HTML5 <audio> element specifies a standard way to embed audio in a web page.

The HTML <audio> Element

To play an audio file in HTML, use the <audio> element:

Example

```
<audio controls>
<source src="horse.ogg" type="audio/ogg">
<source src="horse.mp3" type="audio/mpeg">
Your browser does not support the audio element.
</audio>
```

HTML Audio - How It Works

The controls attribute adds audio controls, like play, pause, and volume. The <source> element allows you to specify alternative audio files which the browser may choose from. The browser will use the first recognized format. The text between the <audio> and </audio> tags will only be displayed in browsers that do not support the <audio> element.

HTML Audio - Media Types

File Format	Media Type
MP3	audio/mpeg
OGG	audio/ogg
WAV	audio/wav

HTML Audio - Methods, Properties, and Events

HTML5 defines DOM methods, properties, and events for the <audio> element. This allows you to load, play, and pause audios, as well as set duration and volume. There are also DOM events that can notify you when an audio begins to play, is paused, etc.

HTML5 Audio Tags

Tag	Description
<audio>	Defines sound content
<source>	Defines multiple media resources for media elements, such as <video> and <audio>

Develop dynamic HTML pages using JavaScript

Objectives : At the end of this lesson you shall be able to

- define function
- explain working of function
- explain the benefit of using function
- explain scope of variables.

JavaScript Functions: A JavaScript function is a block of code designed to perform a particular task.

A JavaScript function is executed when "something" invokes it (calls it).

Example 1

```
function myFunction(p1, p2) {
    return p1 * p2; // the function returns the product of p1
    and p2
}
```

JavaScript Function Syntax

A JavaScript function is defined with the function keyword, followed by a name, followed by parentheses () .

Function names can contain letters, digits, underscores, and dollar signs (same rules as variables).

The parentheses may include parameter names separated by commas: (parameter1, parameter2, ...)

The code to be executed, by the function, is placed inside curly brackets: {}

```
functionName(parameter1, parameter2, parameter3) {
    code to be executed
}
```

Function **parameters** are the names listed in the function definition. Function **arguments** are the real values received by the function when it is invoked. Inside the function, the arguments are used as local variables. A Function is much the same as a **Procedure** or a **Subroutine**, in other programming languages.

Function Invocation

The code inside the function will execute when "something" invokes (calls) the function.

- When an event occurs (when a user clicks a button)
- When it is invoked (called) from JavaScript code
- Automatically (self invoked)

Function Return

When JavaScript reaches a return statement, the function will stop executing. If the function was invoked from a statement, JavaScript will "return" to execute the code after the invoking statement. Functions often compute a return value. The return value is "returned" back to the "caller":

Example 2

Calculate the product of two numbers, and return the result:

```
var x = myFunction(4, 3); // Function is called, return
value will end up in x
```

```
function myFunction(a, b) {
    return a * b; // Function returns the product of a and b }
```

The result in x will be: 12

Why Functions?

You can **reuse** code: Define the code once and use it many times. You can use the same code many times with different arguments, to produce different results.

Example 3

Convert Fahrenheit to Celsius:

```
function toCelsius(fahrenheit) {
    return (5/9) * (fahrenheit-32);
}
```

JavaScript Functions are Objects

In JavaScript, functions are objects. JavaScript functions have properties and methods. You can add your own properties and methods to functions.

JavaScript Functions are Variables Too

In JavaScript, functions can be used as variables:

Example 4

Instead of:

```
temp = toCelsius(32);
```

```
text = "The temperature is " + temp + " Centigrade";
```

You can use:

```
text = "The temperature is " + toCelsius(32) + " Centigrade";
```

JavaScript functions can be redefined like ordinary variables. It can also be passed as values to other functions.

JavaScript Scope

Scope is the set of variables you have access to.

In JavaScript, objects and functions, are also variables. In JavaScript, scope is the set of variables, objects, and functions you have access to. JavaScript has function scope: The scope changes inside functions.

Local JavaScript Variables

Variables declared within a JavaScript function, become LOCAL to the function. Local variables have local scope. They can only be accessed within the function.

Example 5

```
// code here can not use train  
function myFunction() {  
    var train = "Jan Satabdi";  
    // code here can use train  
}  
// code here can use train
```

Since local variables are only recognized inside their functions, variables with the same name can be used in different functions. Local variables are created when a function starts, and deleted when the function is completed.

Global JavaScript Variables

A variable declared outside a function, becomes GLOBAL. A global variable has global scope: All scripts and functions on a web page can access it.

Example 6

```
var train = " Jan Satabdi";  
// code here can use train  
function myFunction() {  
    // code here can use train  
}
```

Automatically Global

If you assign a value to a variable that has not been declared, it will automatically become a GLOBAL variable.

String and number methods in JavaScript

Objectives : At the end of this lesson you shall be able to

- explain string
- explain different string methods.

JavaScript Strings

JavaScript strings are used for storing and manipulating text. A JavaScript string is zero or more characters written inside quotes. You can use single or double quotes.

Example 1

```
var cityname = "Chennai"; // Double quotes  
var cityname = 'Chennai'; // Single quotes
```

You can use quotes inside a string, as long as they don't match the quotes surrounding the string.

Example 2

```
var notes = "You're Welcome";  
var ans = "Coimbatore is called 'Cotton City"';  
var ans = 'Coimbatore is called "Cotton City"';
```

String Length property

This code example will declare train as a global variable, even if it is executed inside a function.

Example 7

```
// code here can use train  
function myFunction() {  
    train = "Jan Satabdi";  
    // code here can use train  
}
```

The Lifetime of JavaScript Variables

The lifetime of a JavaScript variable starts when it is declared. Local variables are deleted when the function is completed. Global variables are deleted when you close the page.

Function Arguments

Function arguments (parameters) work as local variables inside functions.

Global Variables in HTML

With JavaScript, the global scope is the complete JavaScript environment.

In HTML, the global scope is the window object: All global variables belong to the window object.

Example 8

```
// code here can use window.train  
function myFunction() {  
    train = "Jan Satabdi";  
}
```

The length of a string is found in the built in property **length**.

Example 3

```
var s="Computer Operator and Programming Assistant";  
var slen = s.length;
```

Backslash escape character.

The backslash (\) escape character turns special characters into string characters.

Code	Result	Description
\'	'	Single quote
\"	"	Double quote
\\	\	Backslash

The sequence \" inserts a double quote in a string.

Example 4

```
var x = "Bhubaneswar is known as the \"Temple City of India\"";
```

The sequence \' inserts a single quote in a string:

Example 5

```
var x = 'you're Welcome.';
```

The sequence \\ inserts a backslash in a string:

Example 6

```
var x = "The character \\ is called backslash. ";
```

Strings Can be Objects

Normally, JavaScript strings are primitive values, created from literals.

```
var tName = "Veni";
```

But strings can also be defined as objects with the keyword new.

```
var tName = new String("Veni");
```

JavaScript String Methods

String methods help you to work with strings.

Finding a String in a String

indexOf() method

The **indexOf()** method returns the index(position) of the first occurrence of a specified text in a string:

Example 7

```
var str = "When I do good I feel good ";
```

```
var num = str.indexOf("good");
```

Result:

The variable num has the position value 10.

Note: JavaScript counts positions from zero. 0 is the first position in a string, 1 is the second, 2 is the third ...

lastIndexOf() method

The **lastIndexOf()** method returns the index(position) of the last occurrence of a specified text in a string.

Example 8

```
var str = "When I do good I feel good";
```

```
var num = str.lastIndexOf("good");
```

Result:

The variable num has the position value 22.

Both indexOf(), and lastIndexOf() return -1 if the text is not found.

Example 9

```
var str = "When I do good I feel good";
```

```
var num = str.lastIndexOf("better");
```

Result:

The variable num has the position value -1.

Both methods accept a second parameter as the starting position for the search.

Example 10

```
var str = "When I do good I feel good";
```

```
var num = str.indexOf("good", 15);
```

Result:

The variable num has the position value 22.

Searching for a String in a String

The search() method searches a string for a specified value and returns the position of the match.

Example 11

```
var str = "When I do good I feel good";
```

```
var num = str.search("good");
```

Result:

The variable num has the position value 10.

Note: The search() method cannot take a second start position argument.

Extracting String Parts

The slice() Method

slice() extracts a part of a string and returns the extracted part in a new string. The method takes the starting position, and the ending position.

This example slices out a portion of a string from position 14 to position 21.

Example 12

```
var str = "Hockey,Kabadi,Cricket";
```

```
var res = str.slice(14, 21);
```

Result:

The result of res will be Cricket

If a parameter is negative, the position is counted from the end of the string. This example slices out a portion of a string from position -14 to position -8.

Example 13

```
var str = "Hockey,Kabadi,Cricket";
```

```
var res = str.slice(-14, -8);
```

Result:

The result of res will be Kabadi

If you omit the second parameter, the method will slice out the rest of the string.

Example 14

```
var res = str.slice(7);
```

Result:

The result of res will be **Kabadi,Cricket**

The substring() Method

substring() is similar to slice(). The difference is that substring() cannot accept negative indexes.

Example 15

```
var str = "Hockey,Kabadi,Cricket";
var res = str.substring(7, 13);
```

Result:

The result of res will be **Kabadi**

The substr() Method

If you omit the second parameter, substring() will slice out the rest of the string. The substr() Method **substr()** is similar to slice(). The difference is that the second parameter specifies the **length** of the extracted part.

Example 16

```
var str = "Hockey,Kabadi,Cricket";
var res = str.substr(7, 6);
```

Result:

The result of res will be **Kabadi**

If you omit the second parameter, substr() will slice out the rest of the string.

Example 17

```
var str = "Hockey,Kabadi,Cricket";
var res = str.substr(7);
```

Result:

The result of res will be **Kabadi,Cricket**

If the first parameter is negative, the position counts from the end of the string.

Example 18

```
var str = "Hockey,Kabadi,Cricket";
var res = str.substr(-7);
```

Result:

The result of res will be **Cricket**

Replacing String Content

The **replace()** method replaces a specified value with another value in a string.

Example 19

```
str = "I like custard apple";
var newstr = str.replace("custard apple", "mango");
```

Result:

The result of **newstr** will be I like mango

Note : The replace() method does not change the string it is called on. It returns a new string.

By default, the replace() function replaces only the first match.

By default, the replace() function is case sensitive. Writing CUSTARD APPLE (with upper-case) will not work.

To replace case insensitive, use a regular expression with an /i flag (insensitive).

Example 20

```
str = "I like custard apple";
var newstr = str.replace(/CUSTARD APPLE/i, "Mango");
```

Result:

The result of **newstr** will be **I like Mango**

Note that regular expressions are written without quotes.

To replace all matches, use a regular expression with a /g flag (global match).

Example 21

```
str = "I like custard apple and apple";
var newstr = str.replace(/apple/g, "Mango");
```

Result:

The result of **newstr** will be **I like custard Mango and Mango**

Converting to Upper and Lower Case

A string is converted to upper case with **toUpperCase()**

Example 22

```
var str1 = "Information Technology";
var str2 = str1.toUpperCase();
```

Result:

The result of **str2** will be **INFORMATION TECHNOLOGY**

A string is converted to lower case with **toLowerCase()**.

Example 23

```
var str1 = "INFORMATION TECHNOLOGY";
var str2 = str1.toLowerCase();
```

Result:

The result of **str2** will be **information technology**

The concat() Method

concat() joins two or more strings.

Example 24

```
var txt1 = "Mr";
var txt2 = "Selvaraj";
var txt3 = txt1.concat(".", txt2);
```

Result:

The result of **txt3** will be **Mr.Selvaraj**

The **concat()** method can be used instead of the plus operator. These two lines do the same.

Example 25

```
var txt = "Mr" + "." + "Selvaraj";
var txt = "Mr".concat(".", "Selvaraj");
String.trim()
```

`String.trim()` removes whitespace from both sides of a string.

Example 26

```
var str = " India Gate ";
alert(str.trim());
```

Result:

India Gate is displayed without leading and trailing blank spaces in alert box.

Extracting String Characters

The `charAt()` method returns the character at a specified index in a string.

Example 27

```
var str = "Thirumalai Nayakkar Mahal";
str.charAt(0); // returns T
```

The `charCodeAt()` method returns the unicode of the character at a specified index in a string. The method returns an UTF-16 code integer between 0 and 65535.

Example 28

```
var str = "Hill Station";
str.charCodeAt(0); // returns 72
```

Converting a String to an Array

A string can be converted to an array with the `split()` method.

Example 29

```
var txt1 = "Kovai,Nellai,Madurai"; // String
var txt2 = txt1.split(","); // Split on commas
```

Result:

The result of `txt2` will be **Kovai**

If the separator is omitted, the returned array will contain the whole string in index [0]. If the separator is "", the returned array will be an array of single characters.

Example 30

```
var txt = "Anaimai"; // String
txt.split(""); // Split in characters
```

Example 31

```
<!DOCTYPE html>
<html>
<body>
<p id="demo"></p>
<script>
```

```
var str = "Temple";
var arr = str.split("");
var text = "";
var i;
for (i = 0; i < arr.length; i++) {
    text += arr[i] + "<br>"
}
document.getElementById("demo").innerHTML = text;
</script>
</body>
</html>
```

Result:

T
e
m
p
l
e
?

JavaScript Number Methods

Number methods help you work with numbers.

Number Methods and Properties

Primitive values like 2018 or 1.44, cannot have properties and methods because they are not objects.

But with JavaScript, methods and properties are also available to primitive values, because JavaScript treats primitive values as objects when executing methods and properties.

The `toString()` Method

`toString()` returns a number as a string. All number methods can be used on any type of numbers (literals, variables, or expressions).

Example 32

```
var n = 2018;
n.toString(); // returns 2018 from variable x
(2018).toString(); // returns 2018 from literal 2018
(2000+18).toString(); // returns 2018 from expression
2000 + 18
```

The `toExponential()` Method

`toExponential()` returns a string, with a number rounded and written using exponential notation. A parameter defines the number of characters behind the decimal point.

Example 33

```
var x = 3.869;
x.toExponential(2); // returns 3.87e+0
```

```
x.toExponential(4); // returns 3.8690e+0  
x.toExponential(6); // returns 3.869000e+0
```

The parameter is optional. If you don't specify it, JavaScript will not round the number.

The **toFixed()** Method

toFixed() returns a string, with the number written with a specified number of decimals.

Example 34

```
var x = 3.869;  
x.toFixed(0); // returns 4  
x.toFixed(2); // returns 3.87  
x.toFixed(4); // returns 3.8690  
x.toFixed(6); // returns 3.869000
```

Note : toFixed(2) is perfect for working with money.

The **toPrecision()** Method

toPrecision() returns a string, with a number written with a specified length.

Example 35

```
var x = 3.869;  
x.toPrecision(); // returns 3.869  
x.toPrecision(2); // returns 3.9  
x.toPrecision(4); // returns 3.869  
x.toPrecision(6); // returns 3.86900
```

The **valueOf()** Method

valueOf() returns a number as a number.

Example 36

```
var x = 451;  
x.valueOf(); // returns 451 from variable x  
(451).valueOf(); // returns 451 from literal 451  
(400 + 51).valueOf(); // returns 451 from expression 400 + 51
```

Converting Variables to Numbers

There are 3 JavaScript methods that can be used to convert variables to numbers:

- 1 The **Number()** method
- 2 The **parseInt()** method
3. The **parseFloat()** method

These methods are not number methods, but global JavaScript methods.

Global JavaScript Methods

JavaScript global methods can be used on all JavaScript data types. These are the most relevant methods, when working with numbers.

Method	Description
Number()	Returns a number, converted from its argument.
parseInt()	Parses its argument and returns an integer
parseFloat()	Parses its argument and returns a floating point number

The **Number()** Method

Number() can be used to convert JavaScript variables to numbers.

Example 37

```
Number(true); // returns 1  
Number(false); // returns 0  
Number("25"); // returns 25  
Number(" 25 "); // returns 25  
Number("25 "); // returns 25  
Number("25 "); // returns 25  
Number("25.77"); // returns 25.77  
Number("25,77"); // returns NaN  
Number("2577"); // returns NaN  
Number("ITI"); // returns NaN
```

Note : If the number cannot be converted, NaN (Not a Number) is returned.

The **Number()** Method Used on Dates. **Number()** can also convert a date to a number:

Example 38

```
Number(new Date("2018-09-15")); // returns 1536969600000
```

Note : The Number() method above returns the number of milliseconds since 1.1.1970.

The **parseInt()** Method

parseInt() parses a string and returns a whole number. Spaces are allowed. Only the first number is returned.

Example 39

```
parseInt("25"); // returns 25  
parseInt("25.33"); // returns 25  
parseInt("25 20 30"); // returns 25  
parseInt("25 years"); // returns 25  
parseInt("years 25"); // returns NaN
```

Try it yourself "

If the number cannot be converted, **Nan** (Not a Number) is returned.

The **parseFloat()** Method

parseFloat() parses a string and returns a number. Spaces are allowed. Only the first number is returned:

Example 40

```
parseFloat("25");           // returns 25
parseFloat("25.77");        // returns 25.77
parseFloat("255075");        // returns 25
parseFloat("25 years");      // returns 25
parseFloat("years 25");      // returns NaN
```

Note : If the number cannot be converted, NaN (Not a Number) is returned.

?

Number Properties

Property	Description
MIN_VALUE	Returns the smallest number possible in JavaScript
MAX_VALUE	Returns the largest number possible in JavaScript
POSITIVE_INFINITY	Represents infinity (returned on overflow)
NEGATIVE_INFINITY	Represents negative infinity (returned on overflow)
NaN	Represents a "Not-a-Number" value

JavaScript MIN_VALUE and MAX_VALUE

Example 41

```
var n=Number.MAX_VALUE;
```

Result:

MAX_VALUE returns the largest possible number in JavaScript.

1.7976931348623157e+308

Example 42

```
var n=Number.MIN_VALUE;
```

Result:

MIN_VALUE returns the smallest number possible in JavaScript.

Math objects in JavaScript

Objectives : At the end of this lesson you shall be able to

- **explain Math objects in Java Script.**

JavaScript Math Object

The JavaScript Math object allows you to perform mathematical tasks on numbers.

```
Math.PI;           // returns 3.141592653589793
```

Math.round(x) returns the value of x rounded to its nearest integer:

Example 1

```
Math.round(5.8); // returns 6
```

5e-324

JavaScript POSITIVE_INFINITY

Example 43

```
var n=Number.POSITIVE_INFINITY;
POSITIVE_INFINITY is returned on overflow.
```

Example 44

```
var n = 2 / 0;
JavaScript NEGATIVE_INFINITY
```

Example 45

```
var n=Number.NEGATIVE_INFINITY;
NEGATIVE_INFINITY is returned on overflow:
```

Example 46

```
var x = -1 / 0;
JavaScript NaN - Not a Number
```

Example 47

```
var x = Number.NaN;
NaN is a JavaScript reserved word indicating that a number is not a legal number. Trying to do arithmetic with a non-numeric string will result in NaN (Not a Number).
```

Example 48

```
var n = 500 / "Price"; // n will be NaN (Not a Number)
```

Number Properties Cannot be Used on Variables

Number properties belongs to the JavaScript's number object wrapper called **Number**. These properties can only be accessed as **Number.MAX_VALUE**.

Using newNumber.MAX_VALUE, where 'newNumber' is a variable, expression, or value, will return undefined.

Example 49

```
var a = 10;
var b = a.MAX_VALUE; // b becomes undefined
```

Math.round(5.4); // returns 5

Math.pow(x, y) returns the value of x to the power of y:

Example 2

```
Math.pow(5, 2); // returns 25
```

Math.sqrt(x) returns the square root of x:

Example 3

```
Math.sqrt(25); // returns 5
```

Math.abs(x) returns the absolute (positive) value of x:

Example 4

```
Math.abs(-3.5); // returns 3.5
```

Math.ceil(x) returns the value of x rounded up to its nearest integer:

Example 5

```
Math.ceil(5.4); // returns 6
```

Math.floor(x) returns the value of x rounded down to its nearest integer:

Example 6

```
Math.floor(5.8); // returns 5
```

Math.sin(x) returns the sine (a value between -1 and 1) of the angle x (given in radians).

If you want to use degrees instead of radians, you have to convert degrees to radians.

Angle in radians = Angle in degrees x PI / 180.

Example 7

```
Math.sin(90 * Math.PI / 180); // returns 1 (the sine of 90 degrees)
```

```
Math.sin(0 * Math.PI / 180); // returns 0 (the sine of 0 degrees)
```

Math.cos(x) returns the cosine (a value between -1 and 1) of the angle x (given in radians).

If you want to use degrees instead of radians, you have to convert degrees to radians:

Angle in radians = Angle in degrees x PI / 180.

Example 8

```
Math.cos(0 * Math.PI / 180); // returns 1 (the cos of 0 degrees)
```

Math.min() and Math.max() can be used to find the lowest or highest value in a list of arguments:

Example 9

```
Math.min(20,40,6,0,-10); // returns -10
```

Example 10

```
Math.max(20,40,6,0,-10); // returns 40
```

Math.random() returns a random number between 0 (inclusive), and 1 (exclusive):

Example 11

```
Math.random(); // returns a random number
```

Math Properties (Constants)

JavaScript provides 8 mathematical constants that can be accessed with the Math object:

Example 12

```
Math.E // returns Euler's number
```

```
Math.PI // returns PI
```

```
Math.SQRT2 // returns the square root of 2
```

```
Math.SQRT1_2 // returns the square root of 1/2
```

```
Math.LN2 // returns the natural logarithm of 2
```

```
Math.LN10 // returns the natural logarithm of 10
```

```
Math.LOG2E // returns base 2 logarithm of E
```

```
Math.LOG10E // returns base 10 logarithm of E
```

Math Constructor

Unlike other global objects, the Math object has no constructor. Methods and properties are static. All methods and properties (constants) can be used without creating a Math object first.

Math Object Methods

Method	Description
abs(x)	Returns the absolute value of x
acos(x)	Returns the arccosine of x, in radians
asin(x)	Returns the arcsine of x, in radians
atan(x)	Returns the arctangent of x as a numeric value between -PI/2 and PI/2 radians
atan2(y, x)	Returns the arctangent of the quotient of its arguments
ceil(x)	Returns the value of x rounded up to its nearest integer
cos(x)	Returns the cosine of x (x is in radians)
exp(x)	Returns the value of Ex
floor(x)	Returns the value of x rounded down to its nearest integer
log(x)	Returns the natural logarithm (base E) of x
max(x, y, z, ..., n)	Returns the number with the highest value
min(x, y, z, ..., n)	Returns the number with the lowest value
pow(x, y)	Returns the value of x to the power of y

random()	Returns a random number between 0 and 1
round(x)	Returns the value of x rounded to its nearest integer
sin(x)	Returns the sine of x (x is in radians)
sqrt(x)	Returns the square root of x
tan(x)	Returns the tangent of an angle

JavaScript Dates

Objectives : At the end of this lesson you shall be able to

- explain JavaScript Date Objects
- explain JavaScript Date Formats
- explain JavaScript Date get methods
- explain JavaScript Date set methods.

JavaScript Date Objects

By default, JavaScript will use the browser's time zone and display a date as a full text string.

Thu Sep 27 2018 09:09:39 GMT+0530 (India Standard Time)

Creating Date Objects

Date objects are created with the new Date() constructor.

There are 4 ways to create a new date object. They are

- 1 new Date()
- 2 new Date(year, month, day, hours, minutes, seconds, milliseconds)
- 3 new Date(date string)
- 4 new Date(milliseconds)

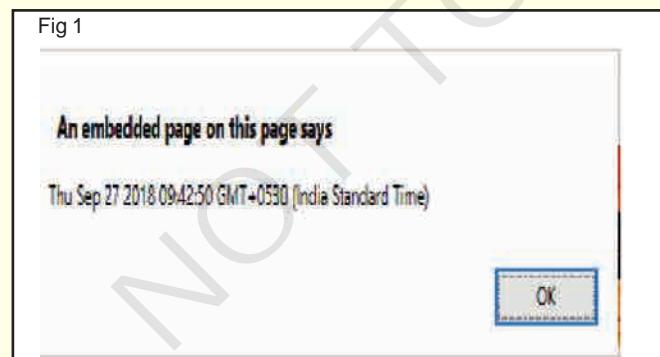
1 new Date()

new Date() creates a new date object with the current date and time.

Example 1:

```
var d = new Date();
alert(d);
```

Result: (Fig 1)



Note Date objects are static. The computer time is ticking, but date objects are not.

2 new Date(year, month, ...)

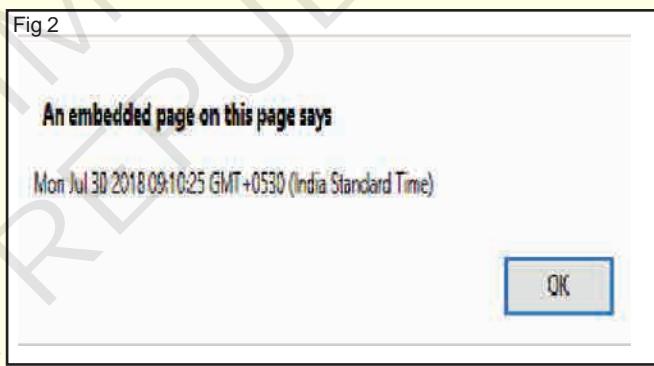
new Date(year, month, ...) creates a new date object with a specified date and time.

Example 2

```
var d = new Date(2018, 06, 30, 09, 10, 25, 0);
```

7 numbers specify year, month, day, hour, minute, second, and millisecond (in that order).

Result: (Fig 2)



Note: JavaScript counts months from 0 to 11. January is 0. December is 11.

6 numbers specify year, month, day, hour, minute, second.

Example 3

```
var d = new Date(2018, 06, 30, 09, 10, 25);
```

5 numbers specify year, month, day, hour, and minute.

Example 4

```
var d = new Date(2018, 06, 30, 09, 10);
```

3 numbers specify year, month, and day.

Example 6

```
var d = new Date(2018, 06, 30);
```

2 numbers specify year and month.

Example 7

```
var d = new Date(2018, 06);
```

Result: (Fig 3)

You cannot omit month. If you supply only one parameter it will be treated as milliseconds.

Fig 3

An embedded page on this page says

Sun Jul 01 2018 00:00:00 GMT+0530 (India Standard Time)

OK

Example 8

```
var d = new Date(2018);
```

Previous Century

One and two digit years will be interpreted as 19xx.

Example 9

```
var d = new Date(96, 04, 12);
```

3 new Date(dateString)

`new Date(dateString)` creates a new date object from a date string.

Example 10

```
var d = new Date("December 20, 2018 10:15:00");
```

Result: (Fig 4)

Fig 4

An embedded page on this page says

Thu Dec 20 2018 10:15:00 GMT+0530 (India Standard Time)

OK

Note: JavaScript stores dates as number of milliseconds since January 01, 1970, 00:00:00 UTC (Universal Time Coordinated). Zero time is January 01, 1970 00:00:00 UTC. Now the time is: 1537962903199 milliseconds past January 01, 1970

4 new Date(milliseconds)

`new Date(milliseconds)` creates a new date object as zero time plus milliseconds.

Example 11

```
var d = new Date(0);
```

01 January 1970 plus 100 000 000 milliseconds is approximately 02 January 1970.

Example 12

```
var d = new Date(100000000);
```

Result: (Fig 5)

January 01 1970 minus 100 000 000 milliseconds is approximately December 31 1969.

Fig 5

An embedded page on this page says

Fri Jan 02 1970 09:16:40 GMT+0530 (India Standard Time)

OK

Example 13

```
var d = new Date(-100000000);
```

Result: (Fig 6)

Fig 6

An embedded page on this page says

Wed Dec 31 1969 01:43:20 GMT+0530 (India Standard Time)

OK

Example 14

```
var d = new Date(86400000);
```

Result:

Fri Jan 02 1970 05:30:00 GMT+0530 (India Standard Time)

Note: Using `new Date(milliseconds)`, creates a new date object as January 1, 1970, 00:00:00 Universal Time (UTC) plus the milliseconds. One day (24 hours) is 86 400 000 milliseconds.

Date Methods

When a Date object is created, a number of methods allow you to operate on it. Date methods allow you to get and set the year, month, day, hour, minute, second, and millisecond of date objects, using either local time or UTC (universal, or GMT) time.

Displaying Dates

By default JavaScript will output dates in full text string format. When you display a date object in HTML, it is automatically converted to a string, with the `toString()` method.

Example 15

```
d = new Date();
```

```
alert(d);
```

Same as:

```
d = new Date();
```

```
alert(d.toString());  
The toUTCString() method converts a date to a UTC string.
```

Example 16

```
var d = new Date();  
alert(d);
```

Result: (Fig 7)

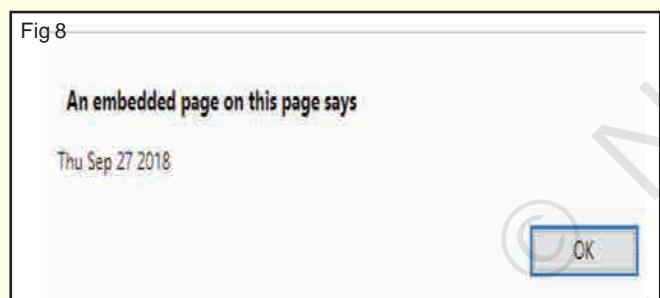
The toDateString() method converts a date to a more readable format.



Example 17

```
var d = new Date();  
alert(d.toDateString());
```

Result: (Fig 8)



JavaScript Date Formats

JavaScript Date Input

There are generally 3 types of JavaScript date input formats.

Type	Example
ISO Date	"2002-06-30" (The International Standard)
Short Date	"06/30/2002"
Long Date	"Jun 30 2002" or "30 Jun 2002"

JavaScript Date Output

Independent of input format, JavaScript will output dates in full text string format.

JavaScript ISO Dates

ISO 8601 is the international standard for the representation of dates and times. The ISO 8601 syntax (YYYY-MM-DD) is also the preferred JavaScript date format.

Example 18 (Complete date)

```
var d = new Date("2002-06-30");
```

Note: The computed date will be relative to your time zone. Depending on your time zone, the result above will vary between June 29 and June 30.

ISO Dates (Year and Month)

ISO dates can be written without specifying the day (YYYY-MM).

Example 19

```
var d = new Date("2002-06");
```

Result:

Sat Jun 01 2002 05:30:00 GMT+0530 (India Standard Time)

ISO Dates (Only Year)

ISO dates can be written without month and day (YYYY).

Example 20

```
var d = new Date("2011");
```

Result:

Sat Jan 01 2011 05:30:00 GMT+0530 (India Standard Time)

ISO Dates (Date-Time)

ISO dates can be written with added hours, minutes, and seconds (YYYY-MM-DDTHH:MM:SSZ)

Example 21

```
var d = new Date("2011-12-20T12:00:00Z");
```

Result:

Tue Dec 20 2011 17:30:00 GMT+0530 (India Standard Time)

Note: Date and time is separated with a capital T. UTC time is defined with a capital letter Z.

JavaScript Short Dates.

Short dates are written with an "MM/DD/YYYY" syntax like this.

Example 22

```
var d = new Date("06/30/2002");
```

Result:

Sun Jun 30 2002 00:00:00 GMT+0530 (India Standard Time)

Note: In some browsers, months or days with no leading zeroes may produce an error. The behavior of "YYYY/MM/DD" is undefined. Some browsers will try to guess the format. Some will return NaN. The behavior of "DD-MM-YYYY" is also undefined. Some browsers will try to guess the format. Some will return NaN.

JavaScript Long Dates

Long dates are most often written with a "MMM DD YYYY" syntax like this.

Example 23

```
var d = new Date("Aug 31 2012");
```

Month and day can be in any order.

Example 24

```
var d = new Date("31 Aug 2012");
```

And, month can be written in full (January), or abbreviated (Jan).

Example 25

```
var d = new Date("August 31 2012");
```

```
var d = new Date("AUGUST 31 2012");
```

Note: Commas are ignored. Names are case insensitive.

Date Input - Parsing Dates

If you have a valid date string, you can use the Date.parse() method to convert it to milliseconds. Date.parse() returns the number of milliseconds between the date and January 1, 1970.

Example 26

```
var msec = Date.parse("Sep 15, 1996");
```

```
document.getElementById("demo").innerHTML = msec;
```

Result:

842725800000

You can then use the number of milliseconds to convert it to a date object.

Example 27

```
var msec = Date.parse("Sep 15, 1996");
```

```
var d = new Date(msec);
```

```
document.getElementById("demo").innerHTML = d;
```

Result:

Sun Sep 15 1996 00:00:00 GMT+0530 (India Standard Time)

JavaScript Get Date Methods

These methods can be used for getting information from a date object.

Method	Description
getFullYear()	Get the year as a four digit number (yyyy)
getMonth()	Get the month as a number (0-11)
getDate()	Get the day as a number (1-31)
getHours()	Get the hour (0-23)
getMinutes()	Get the minute (0-59)
getSeconds()	Get the second (0-59)
getMilliseconds()	Get the millisecond (0-999)
getTime()	Get the time (milliseconds since January 1, 1970)

getDay()	Get the weekday as a number (0-6)
Date.now()	Get the time. ECMAScript 5.

The getTime() Method

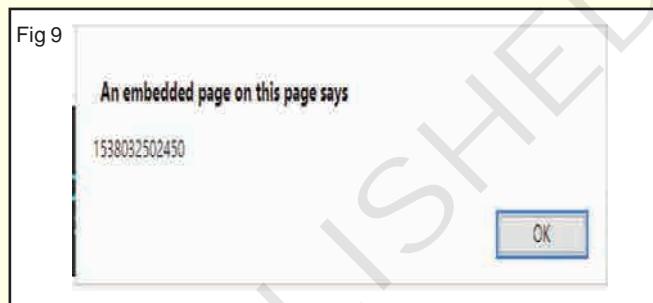
The getTime() method returns the number of milliseconds since January 1, 1970.

Example 28

```
var d = new Date();
```

```
alert(d.getTime());
```

Result: (Fig 9)



The getFullYear() Method

The getFullYear() method returns the year of a date as a four digit number.

Example 29

```
var d = new Date();
```

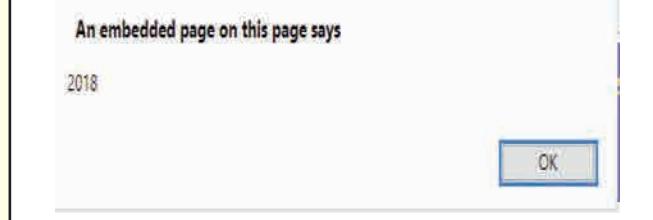
```
alert(d.getFullYear());
```

Result: (Fig 10)

The getMonth() Method

The getMonth() method returns the month of a date as a number (0-11).

Fig 10



Example 30

```
var d = new Date();
```

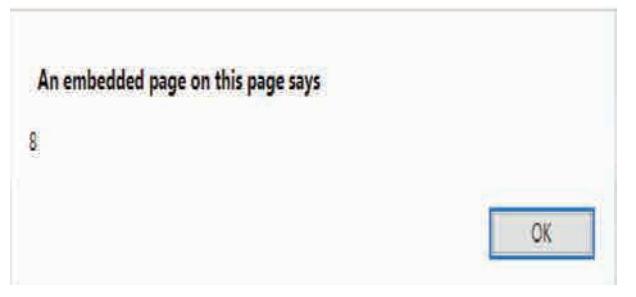
```
document.getElementById("demo").innerHTML = d.getMonth();
```

Result: (Fig 11)

In JavaScript, the first month (January) is month number 0, so December returns month number 11.

You can use an array of names, and getMonth() to return the month as a name.

Fig 11



Example 31

```
var d = new Date();
var months = ["January", "February", "March", "April",
"May", "June", "July", "August", "September", "October",
"November", "December"];
document.getElementById("demo").innerHTML =
months[d.getMonth()];
```

Result: (Fig 12)

The getDate() Method

The getDate() method returns the day of a date as a number (1-31).

Fig 12



Example 32

```
var d = new Date();
alert(d.getDate());
```

Result: (Fig 13)

The getHours() Method

The getHours() method returns the hours of a date as a number (0-23).

Fig 13



Example 33

```
var d = new Date();
alert(d.getHours());
```

Result: (Fig 14)

The getMinutes() Method

The getMinutes() method returns the minutes of a date as a number (0-59).

Fig 14



Example 34

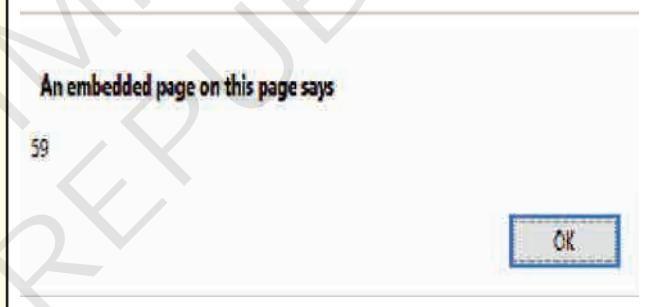
```
var d = new Date();
alert(d.getMinutes());
```

Result: (Fig 15)

The getSeconds() Method

The getSeconds() method returns the seconds of a date as a number (0-59).

Fig 15



Example 35

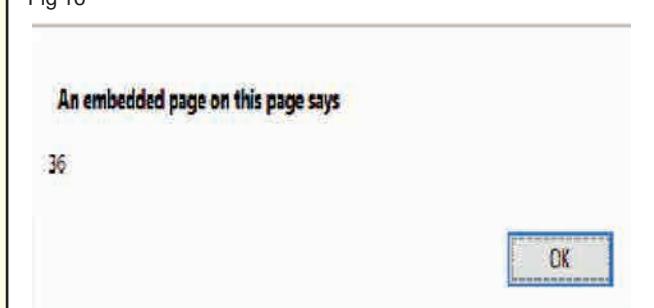
```
var d = new Date();
alert(d.getSeconds());
```

Result: (Fig 16)

The getMilliseconds() Method

The getMilliseconds() method returns the milliseconds of a date as a number (0-999).

Fig 16



Example 36

```
var d = new Date();
alert(d.getMilliseconds());
```

Result: (Fig 17)

The getDay() Method

The getDay() method returns the weekday of a date as a number (0-6).

Fig 17



Example 37

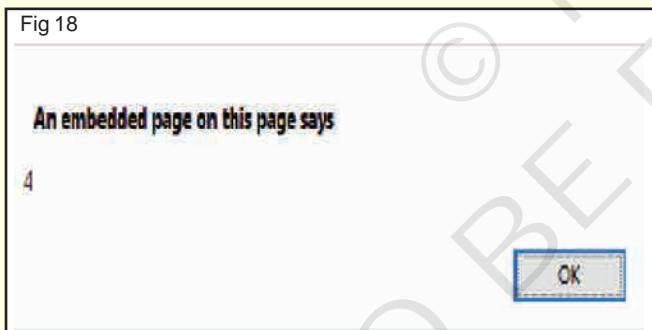
```
var d = new Date();
alert(d.getDay());
```

Result: (Fig 18)

In JavaScript, the first day of the week (0) means "Sunday", even if some countries in the world consider the first day of the week to be "Monday".

You can use an array of names, and getDay() to return the weekday as a name.

Fig 18

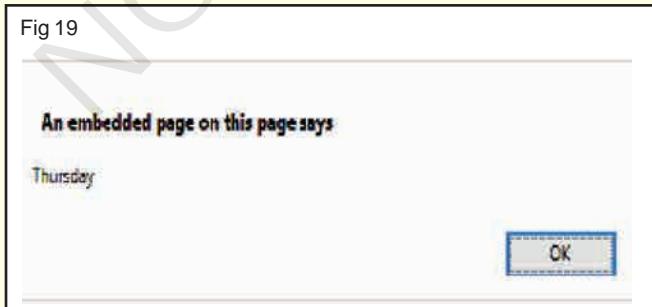


Example 38

```
var d = new Date();
var days = ["Sunday", "Monday", "Tuesday",
           "Wednesday", "Thursday", "Friday", "Saturday"];
alert(days[d.getDay()]);
```

Result: (Fig 19)

Fig 19



UTC Date Methods

UTC date methods are used for working with UTC dates (Universal Time Zone dates).

Method	Description
getUTCDate()	Same as getDate(), but returns the UTC date
getUTCDay()	Same as getDay(), but returns the UTC day
getUTCFullYear()	Same as getFullYear(), but returns the UTC year
getUTCHours()	Same as getHours(), but returns the UTC hour
getUTCMilliseconds()	Same as getMilliseconds(), but returns the UTC milliseconds
getUTCMilliseconds()	Same as getMinutes(), but returns the UTC minutes
getUTCMonth()	Same as getMonth(), but returns the UTC month
getUTCSeconds()	Same as getSeconds(), but returns the UTC seconds

JavaScript Set Date Methods

Set Date methods let you set date values (years, months, days, hours, minutes, seconds, milliseconds) for a Date Object.

Set Date Methods

Set Date methods are used for setting a part of a date.

Method	Description
setDate()	Set the day as a number (1-31)
setFullYear()	Set the year (optionally month and day)
setHours()	Set the hour (0-23)
setMilliseconds()	Set the milliseconds (0-999)
setMinutes()	Set the minutes (0-59)
setMonth()	Set the month (0-11)
setSeconds()	Set the seconds (0-59)
setTime()	Set the time (milliseconds since January 1, 1970)

The setFullYear() Method

The setFullYear() method sets the year of a date object. In this example to 2020.

Example 39

```
<script>
var d = new Date();
d.setFullYear(2020);
alert(d);
</script>
```

Result: (Fig 20)

Fig 20



Note: The `setFullYear()` method can optionally set month and day.

Example 40

```
<script>
var d = new Date();
d.setFullYear(2018, 10, 2);
alert(d);
<script>
```

Result: (Fig 21)

Fig 21



Note: month counts from 0. December is month 11

The `setMonth()` Method

The `setMonth()` method sets the month of a date object (0-11).

Example 41

```
<script>
var d = new Date();
d.setMonth(2);
alert(d);
</script>
```

Result: (Fig 22)

The `setDate()` Method

The `setDate()` method sets the day of a date object (1-31).

Example 42

```
<script>
var d = new Date();
d.setDate(18);
```

Fig 22



alert(d);

</script>

Result: (Fig 23)

The `setDate()` method can also be used to add days to a date.

Fig 23



Example 43

```
<script>
var d = new Date();
d.setDate(d.getDate() + 25);
alert(d);
</script>
```

Result: (Fig 24)

If adding days, shifts the month or year, the changes are handled automatically by the Date object.

Fig 24



The `setHours()` Method

The `setHours()` method sets the hours of a date object (0-23).

Example 44

```
<script>
var d = new Date();
```

```
d.setHours(20);
```

```
alert(d);
```

```
</script>
```

Result: (Fig 25)

The setMinutes() Method

The setMinutes() method sets the minutes of a date object (0-59).

Fig 25



OK

Example 45

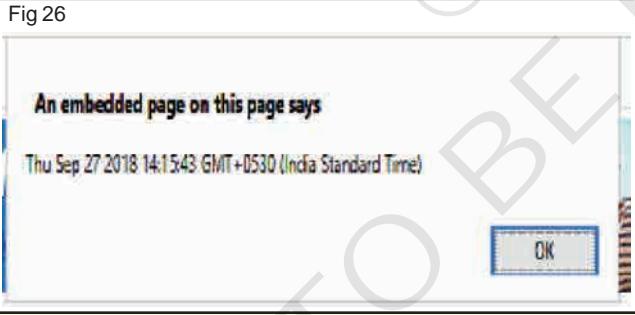
```
<script>  
var d = new Date();  
d.setMinutes(15);  
alert(d);  
</script>
```

Result: (Fig 26)

The setSeconds() Method

The setSeconds() method sets the seconds of a date object (0-59).

Fig 26



OK

Example 46

```
<script>  
var d = new Date();  
d.setSeconds(20);  
alert(d);  
</script>
```

Result: (Fig 27)

Compare Dates

Dates can easily be compared.

The following example compares today's date with January 14, 2100.

Fig 27



OK

Example 47

```
var date1 = new Date(2010, 00, 15); //Year, Month, Date  
var date2 = new Date(2011, 00, 15); //Year, Month, Date  
if(date1 > date2)  
{  
    alert("Date One is greater than Date Two.");  
}  
else  
{  
    alert("Date Two is greater than Date One.");  
}
```

Deploy web project using IIS

Objectives : At the end of this lesson you shall be able to

- define DOM
- explain DOM methods
- explain DOM documents
- describe HTML DOM elements
- explain HTML DOM events
- know Open Source Software

JavaScript HTML DOM

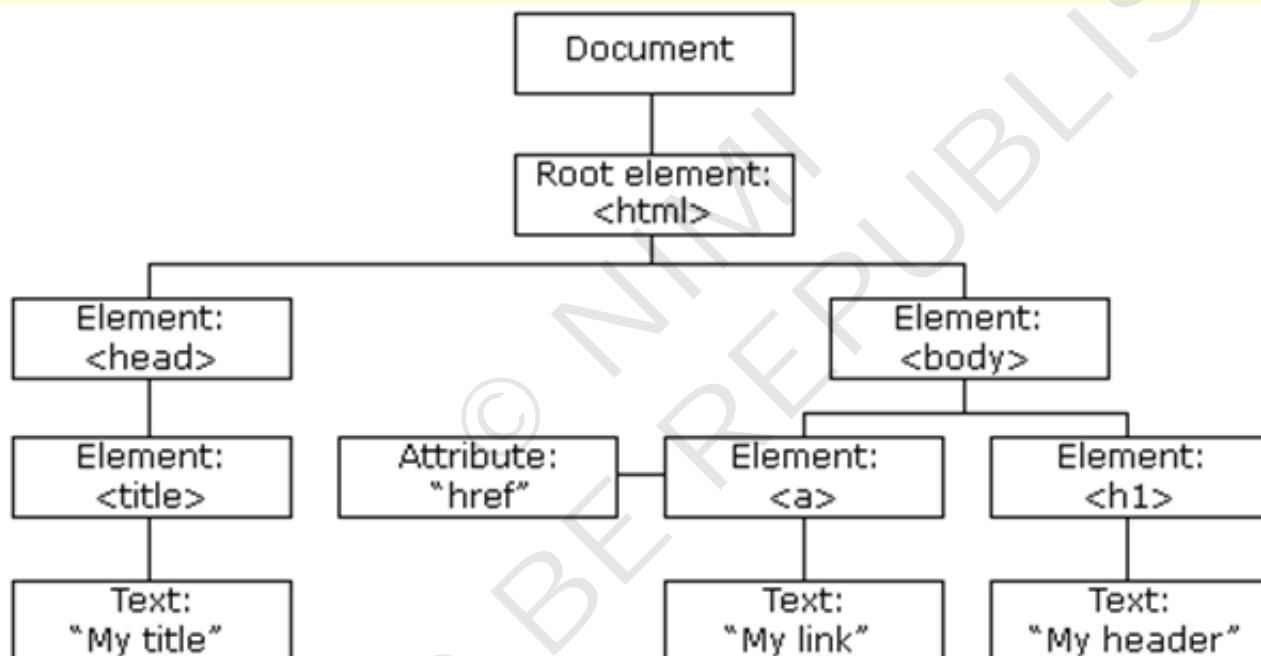
With the HTML DOM, JavaScript can access and modify all the elements of an HTML document.

The HTML DOM (Document Object Model): When a web page is loaded, the browser creates a Document Object Model of the page.

The HTML DOM model is constructed as a tree of Objects:

The HTML DOM Tree of Objects

With the object model, JavaScript gets all the power it needs to create dynamic HTML:



- JavaScript can change all the HTML elements in the page.
- JavaScript can change all the HTML attributes in the page
- JavaScript can change all the CSS styles in the page
- JavaScript can remove existing HTML elements and attributes
- JavaScript can add new HTML elements and attributes
- JavaScript can react to all existing HTML events in the page

DOM

The DOM is a W3C (World Wide Web Consortium) standard.

The DOM defines a standard for accessing documents:

"The W3C Document Object Model (DOM) is a platform and language-neutral interface that allows programs and scripts to dynamically access and update the content, structure, and style of a document."

The W3C DOM standard is separated into 3 different parts:

- Core DOM - standard model for all document types
- XML DOM - standard model for XML documents
- HTML DOM - standard model for HTML documents

What is the HTML DOM?

The HTML DOM is a standard object model and programming interface for HTML. It defines:

- The HTML elements as objects
- The properties of all HTML elements
- The methods to access all HTML elements
- The events for all HTML elements

In other words: The HTML DOM is a standard for how to get, change, add, or delete HTML elements.

HTML DOM Methods

HTML DOM methods are **actions** you can perform (on HTML Elements)

HTML DOM properties are **values** (of HTML Elements) that you can set or change.

The DOM Programming Interface

The HTML DOM can be accessed with JavaScript (and with other programming languages).

In the DOM, all HTML elements are defined as **objects**. The **programming interface** is the properties and methods of each object. A **property** is a value that you can get or set (like changing the content of an HTML element). A **method** is an action you can do (like add or deleting an HTML element).

Example 1

The following example changes the content (the innerHTML) of the <p> element with id="demo":

```
<html>
<body>
<p id="demo"></p>
<script>
document.getElementById("demo").innerHTML =
"Welcome to JavaScript!";
```

Finding HTML Elements

Method	Description
document.getElementById()	Find an element by element id
document.getElementsByTagName()	Find elements by tag name
document.getElementsByClassName()	Find elements by class name

Changing HTML Elements

Method	Description
element.innerHTML=	Change the inner HTML of an element
element.attribute=	Change the attribute of an HTML element
element.setAttribute(attribute,value)	Change the attribute of an HTML element
element.style.property=	Change the style of an HTML element

Adding and Deleting HTML Elements

Method	Description
document.createElement()	Create an HTML element
document.removeChild()	Remove an HTML element
document.appendChild()	Add an HTML element
document.replaceChild()	Replace an HTML element
document.write(text)	Write into the HTML output stream

```
</script>
```

```
</body>
```

```
</html>
```

In the example above, getElementById is a method, while innerHTML is a property.

The getElementById Method

The most common way to access an HTML element is to use the id of the element. In the example above the getElementById method used **id="demo"** to find the element.

The innerHTML Property

The easiest way to get the content of an element is by using the innerHTML property. The innerHTML property is useful for getting or replacing the content of HTML elements.

Note: The innerHTML property can be used to get or change any HTML element, including <html> and <body>.

HTML DOM Document

HTML DOM document object

The document object is the owner of all other objects in your web page. In the HTML DOM object model, the document object represents your web page. If you want to access objects in an HTML page, you always start with accessing the document object.

Below are some examples of how you can use the document object to access and manipulate HTML.

Adding Events handlers

Method	Description
document.getElementById(id).onclick=function(){code}	Adding event handler code to an onclick event

JavaScript HTML DOM Elements

Finding HTML Elements

Often, with JavaScript, you want to manipulate HTML elements.

To do so, you have to find the elements first. There are a couple of ways to do this:

- Finding HTML elements by id
- Finding HTML elements by tag name
- Finding HTML elements by class name
- Finding HTML elements by HTML object collections

Finding HTML Elements by Id

The easiest way to find HTML elements in the DOM, is by using the element id.

This example finds the element with id="demo":

Example 2

```
var x = document.getElementById("demo");
```

If the element is found, the method will return the element as an object (in x).

If the element is not found, x will contain null.

Finding HTML Elements by Tag Name

This example finds the element with id="main", and then finds all <p> elements inside "main":

Example 3

```
var x = document.getElementById("main");
var y = x.getElementsByTagName("p");
```

Finding HTML Elements by Class Name

If you want to find all HTML elements with the same class name, use this method getElementsByClassName()

Example 4

```
document.getElementsByClassName("intro");
```

The example above returns a list of all elements with class="intro".

Note: Finding elements by class name does not work in Internet Explorer 5,6,7, and 8.

Finding HTML Elements by HTML Object Collections

This example finds the form element with id="frm1", in the forms collection, and displays all element values:

Method	Description	DOM
document.anchors	Returns all <a> with a value in the name attribute	1
document.applets	Returns all <applet> elements (Deprecated in HTML5)	1
document.baseURI	Returns the absolute base URI of the document	3
document.body	Returns the <body> element	1
document.cookie	Returns the document's cookie	1
document.doctype	Returns the document's doctype	3
document.documentElement	Returns the <html> element	3
document.documentElement	Returns the mode used by the browser	3
document.documentElement	Returns the URI of the document	3
document.domain	Returns the domain name of the document server	1
document.domConfig	Returns the DOM configuration	3
document.embeds	Returns all <embed> elements	3
document.forms	Returns all <form> elements	1
document.head	Returns the <head> element	3
document.images	Returns all <image> elements	1
document.implementation	Returns the DOM implementation	3
document.inputEncoding	Returns the document's encoding (character set)	3
document.lastModified	Returns the date and time the document was updated	3
document.links	Returns all <area> and <a> elements value in href	1

document.readyState	Returns the (loading) status of the document	3
document.referrer	Returns the URI of the referrer (the linking document)	1
document.scripts	Returns all <script> elements	3
document.strictErrorChecking	Returns if error checking is enforced	3
document.title	Returns the <title> element	1
document.URL	Returns the complete URL of the document	1

Example 5

```
var x = document.getElementById("frm1");
var text = "";
var i;
for (i = 0; i < x.length; i++) {
    text += x.elements[i].value + "<br>";
}
document.getElementById("demo").innerHTML = text;
```

The following HTML objects (and object collections) are also accessible:

- document.anchors
- document.body
- document.documentElement
- document.embeds
- document.forms
- document.head
- document.images
- document.links
- document.scripts
- document.title

The HTML DOM allows JavaScript to change the content of HTML elements.

Changing the HTML Output Stream

JavaScript can create dynamic HTML content. In JavaScript, document.write() can be used to write directly to the HTML output stream.

Example 6

```
<!DOCTYPE html>
<html>
<body>
<script>
document.write(Date());
</script>
</body>
</html>
```

Note: Never use document.write() after the document is loaded. It will overwrite the document.

Changing HTML Content

The easiest way to modify the content of an HTML element is by using the innerHTML property. To change the content of an HTML element, use this syntax.

```
document.getElementById(id).innerHTML = new HTML
```

This example changes the content of a <p> element:

Example 7

```
<html>
<body>
<p id="p1">Hello World!</p>
<script>
document.getElementById("p1").innerHTML = "New text!";
</script>
</body>
</html>
```

Example explained:

- The HTML document above contains a <p> element with id="p1"
- We use the HTML DOM to get the element with id="p1"
- A JavaScript changes the content (innerHTML) of that element to "New text!"

This example changes the content of an <h1> element:

Example 8

```
<!DOCTYPE html>
<html>
<body>
<h1 id="id1">Old Heading</h1>
<script>
var element = document.getElementById("id1");
element.innerHTML = "New Heading";
</script>
</body>
</html>
```

Example explained:

- The HTML document above contains an `<h1>` element with `id="id1"`
- We use the HTML DOM to get the element with `id="id1"`
- A JavaScript changes the content (`innerHTML`) of that element to "New Heading"

Changing the Value of an Attribute

To change the value of an HTML attribute, use this syntax.

```
document.getElementById(id).attribute = new value
```

This example changes the value of the `src` attribute of an `` element.

Example 9

```
<!DOCTYPE html>
<html>
<body>

<script>
document.getElementById("Image1").src      =
"newflower.jpg";
</script>
</body>
</html>
```

Example explained:

- The HTML document above contains an `` element with `id="myImage"`
- We use the HTML DOM to get the element with `id="myImage"`
- A JavaScript changes the `src` attribute of that element from "smiley.gif" to "landscape.jpg"

Changing HTML Style

To change the style of an HTML element, use this syntax.

```
document.getElementById(id).style.property = new style
```

The following example changes the style of a `<p>` element:

Example 10

```
<html>
<body>
<p id="p2">Hello World!</p>
<script>
document.getElementById("p2").style.color = "green";
</script>
<p>The paragraph above was changed by a script.</p>
</body>
</html>
```

Using Events: The HTML DOM allows you to execute code when an event occurs. Events are generated by the browser when "things happen" to HTML elements.

- An element is clicked on
- The page has loaded
- Input fields are changed

This example changes the style of the HTML element with `id="id1"`, when the user clicks a button.

Example 11

```
<!DOCTYPE html>
<html>
<body>
<h1 id="id1">Heading1</h1>
<button type="button"
onclick="document.getElementById('id1').style.color =
'blue'">
Click Me</button>
</body>
</html>
```

JavaScript HTML DOM Events

HTML DOM allows JavaScript to react to HTML events.

Reacting to Events

A JavaScript can be executed when an event occurs, like when a user clicks on an HTML element. To execute code when a user clicks on an element, add JavaScript code to an HTML event attribute

`onclick=JavaScript`

Examples of HTML events:

- When a user clicks the mouse
- When a web page has loaded
- When an image has been loaded
- When the mouse moves over an element
- When an input field is changed
- When an HTML form is submitted
- When a user strokes a key

In this example, the content of the `<h1>` element is changed when a user clicks on it.

Example 12

```
<!DOCTYPE html>
<html>
<body>
<h1 onclick="this.innerHTML = 'Ooops!'">Click on this
text!</h1>
</body>
</html>
```

HTML Event Attributes

To assign events to HTML elements you can use event attributes.

Example 13

Assign an onclick event to a button element:

```
<button onclick="displayDate()">Try it</button>
```

In the example above, a function named displayDate will be executed when the button is clicked.

Assign Events Using the HTML DOM

The HTML DOM allows you to assign events to HTML elements using JavaScript.

Example 14

Assign an onclick event to a button element:

```
<script>
document.getElementById("myBtn").onclick = displayDate;
</script>
```

In the example above, a function named displayDate is assigned to an HTML element with the id="myBtn".

The function will be executed when the button is clicked.

The onload and onunload Events

The onload and onunload events are triggered when the user enters or leaves the page. The onload event can be used to check the visitor's browser type and browser version and load the proper version of the web page based on the information. The onload and onunload events can be used to deal with cookies.

Example 15

```
<body onload="checkCookies()">
```

The onchange Event

The onchange event is often used in combination with validation of input fields. Below is an example of how to use the onchange. The uppercase() function will be called when a user changes the content of an input field.

Example 16

```
<input type="text" id="fname" onchange="uppercase()">
```

The onmouseover and onmouseout Events

The onmouseover and onmouseout events can be used to trigger a function when the user mouses over or out of, an HTML element.

The onmousedown, onmouseup and onclick Events

The onmousedown, onmouseup and onclick events are all parts of a mouse-click. First when a mouse-button is clicked, the onmousedown event is triggered, then, when the mouse-button is released, the onmouseup event is triggered, finally, when the mouse-click is completed, the onclick event is triggered.

onmousedown and onmouseup

Change an image when a user holds down the mouse button.

onload

Display an alert box when the page has finished loading.

onfocus

Change the background-color of an input field when it gets focus.

Mouse Events

Change the color of an element when the cursor moves over it.

DOM Event Listener

The addEventListener() method

Add an event listener that fires when a user clicks a button.

Example 17

```
<!DOCTYPE html>
<html>
<body>
<h2>JavaScript addEventListener()</h2>
<p>This example uses the addEventListener() method to attach a click event to a button.</p>
<button id="myBtn">Try it</button>
<p id="demo"></p>
<script>
document.getElementById("myBtn").addEventListener("click", displayDate);
function displayDate() {
document.getElementById("demo").innerHTML = Date();
}
</script>
</body>
</html>
```

Result: (Fig 1)

Fig 1

JavaScript addEventListener()

This example uses the addEventListener() method to attach a click event to a button.



Thu Sep 27 2018 17:10:55 GMT-0530 (India Standard Time)

- The addEventListener() method attaches an event handler to the specified element.
- The addEventListener() method attaches an event handler to an element without overwriting existing event handlers.

- You can add many event handlers to one element.
- You can add many event handlers of the same type to one element, i.e two "click" events.
- You can add event listeners to any DOM object not only HTML elements. i.e the window object.
- The `addEventListener()` method makes it easier to control how the event reacts to bubbling.
- When using the `addEventListener()` method, the JavaScript is separated from the HTML markup, for better readability and allows you to add event listeners even when you do not control the HTML markup.
- You can easily remove an event listener by using the `removeEventListener()` method.

Open Source Software

You can build a website using these popular free and open source website building tools. Nowadays, whether you are an individual entrepreneur or representing a business organisation, a website is a must for personal and professional growth. Organisations are spending lots of money to build attractive websites. The following are some of the open source website building tools that you can use to build your website on your own, without much knowledge about programming or the Internet.

1 WordPress

The official websites for WordPress are
<https://wordpress.com> and
<https://wordpress.org/>.

2 Kompozer

The official website for Kompozer is
<https://www.kompozer.net>.

3 Joomla

The official website for Joomla is
<https://www.joomla.org/>.

4 Drupal

The official website for Drupal is
<https://www.drupal.org/>.

5 OpenCms

The official website for OpenCms is
<http://www.opencms.org/en/>.

Create advanced formulas and macros

Objectives: At the end of this lesson you shall be able to

- types of references
- custom number format

Advanced Excel

Types of references

There are two types of cell references: relative and absolute. Relative and absolute references behave differently when copied and filled to other cells. Relative references change when a formula is copied to another cell. Absolute references, on the other hand, remain constant no matter where they are copied.

Relative references

By default, all cell references are relative references. When copied across multiple cells, they change based on the relative position of rows and columns. For example, if you copy the formula =A1+B1 from row 1 to row 2, the formula will become =A2+B2. Relative references are especially convenient whenever you need to repeat the same calculation across multiple rows or columns.

Absolute references

There may be times when you do not want a cell reference to change when filling cells. Unlike relative references, absolute references do not change when copied or filled. You can use an absolute reference to keep a row and/or column constant.

An absolute reference is designated in a formula by the addition of a dollar sign (\$) before the column and row. If it precedes the column or row (but not both), it's known as a mixed reference.

\$A\$2	The column and th row do not change when copied
A\$2	The row does not change when copied
\$A2	The column does not change when copied

You will use the relative (A2) and absolute (\$A\$2) formats in most formulas. Mixed references are used less frequently.

When writing a formula in Microsoft Excel, you can press the F4 key on your keyboard to switch between relative, absolute, and mixed cell references, as shown in the video below. This is an easy way to quickly insert an absolute reference.

Excel Linkage custom Format and Excel Protection

If a built-in number format does not meet your needs, you can create a new number format that is based on an existing number format and add it to the list of custom

number formats. For example, if you're creating a spreadsheet that contains customer information, you can create a number format for telephone numbers. You can apply the custom number format to a string of numbers in a cell to format them as a telephone number.

Important: Custom number formats affect only the way a number is displayed and do not affect the underlying value of the number. Custom number formats are stored in the active workbook and are not available to new workbooks that you open.

Create a custom number format

- 1 On the Home tab, in the Number group, click More Number Formats at the bottom of the Number Format list .
- 2 In the Format Cells dialog box, under Category, click Custom.
- 3 In the Type list, select the built-in format that most resembles the one that you want to create. For example, 0.00.
The number format that you select appears in the Type box.
- 4 In the Type box, modify the number format codes to create the exact format that you want. For example, 000-000-0000.
Your changes will not alter the built-in format. Instead, your changes create a new custom number format.
- 5 When you have finished, click OK.

Apply a custom number format

- 1 Select the cell or range of cells that you want to format.
- 2 On the Home tab, in the Number group, click More Number Formats at the bottom of the Number Format list .
- 3 In the Format Cells dialog box, under Category, click Custom.
- 4 At the bottom of the Type list, select the built-in format that you just created. For example, 000-000-0000.
The number format that you select appears in the Type box.
- 5 Click OK.

Delete a custom number format

- 1 On the Home tab, in the Number group, click More Number Formats at the bottom of the Number Format list .
- 2 In the Format Cells dialog box, under Category, click Custom.
- 3 In the Type list, select the custom number format, and then click Delete.

Notes

- **Built-in number formats cannot be deleted.**
- **Any cells in the workbook that were formatted with the deleted custom format will be displayed in the default General format.**

Excel Protection

To prevent others from accessing data in your Excel files, protect your Excel file with a password.

Note: This topic covers file-level protection only, and not workbook or worksheet protection. To learn the difference between protecting your Excel file, workbook, or a worksheet, see Protection and security in Excel.

- 1 Select File > Info.
- 2 Select the Protect Workbook box and choose Encrypt with Password.
- 3 Enter a password in the Password box, and then select OK.
- 4 Confirm the password in the Reenter Password box, and then select OK.

Warning

- Microsoft cannot retrieve forgotten passwords, so be sure that your password is especially memorable.
- There are no restrictions on the passwords you use with regards to length, characters or numbers, but passwords are case-sensitive.
- It's not always secure to distribute password-protected files that contain sensitive information such as credit card numbers.
- Be cautious when sharing files or passwords with other users. You still run the risk of passwords them falling into the hands of unintended users. Remember that locking a file with a password does not necessarily protect your file from malicious intent.

Excel Tips

- 1 Use Pivot tables to recognize and make sense of data.
- 2 Add more than one row or column.
- 3 Use filters to simplify your data.

- 4 Remove duplicate data points or sets.
- 5 Transpose rows into columns.
- 6 Split up text information between columns.
- 7 Use these formulas for simple calculations.
- 8 Get the average of numbers in your cells.
- 9 Use conditional formatting to make cells automatically change color based on data.
- 10 Use IF Excel formula to automate certain Excel functions.
- 11 Use dollar signs to keep one cell's formula the same regardless of where it moves.
- 12 Use the VLOOKUP function to pull data from one area of a sheet to another.
- 13 Use INDEX and MATCH formulas to pull data from horizontal columns.
- 14 Use the COUNTIF function to make Excel count words or numbers in any range of cells.
- 15 Combine cells using ampersand.
- 16 Add checkboxes.
- 17 Hyperlink a cell to a website.
- 18 Add drop-down menus.
- 19 Use the format painter.

Pivot Table

You can use a PivotTable to summarize, analyze, explore, and present summary data. PivotCharts complement PivotTables by adding visualizations to the summary data in a PivotTable, and allow you to easily see comparisons, patterns, and trends. Both PivotTables and PivotCharts enable you to make informed decisions about critical data in your enterprise. You can also connect to external data sources such as SQL Server tables, SQL Server Analysis Services cubes, Azure Marketplace, Office Data Connection (.odc) files, XML files, Access databases, and text files to create PivotTables, or use existing PivotTables to create new tables.

A PivotTable is an interactive way to quickly summarize large amounts of data. You can use a PivotTable to analyze numerical data in detail, and answer unanticipated questions about your data. A PivotTable is especially designed for:

- Querying large amounts of data in many user-friendly ways.
- Subtotaling and aggregating numeric data, summarizing data by categories and subcategories, and creating custom calculations and formulas.
- Expanding and collapsing levels of data to focus your results, and drilling down to details from the summary data for areas of interest to you.

- Moving rows to columns or columns to rows (or "pivoting") to see different summaries of the source data.
- Filtering, sorting, grouping, and conditionally formatting the most useful and interesting subset of data enabling you to focus on just the information you want.
- Presenting concise, attractive, and annotated online or printed reports.

For example, here's a simple list of household expenses on the left, and a PivotTable based on the list to the right:

Ways to work with pivot table

After you create a PivotTable by selecting its data source, arranging fields in the PivotTable Field List, and choosing an initial layout, you can perform the following tasks as you work with a PivotTable:

Explore the data by doing the following

- Expand and collapse data, and show the underlying details that pertain to the values.
- Sort, filter, and group fields and items.
- Change summary functions, and add custom calculations and formulas.

Change the form layout and field arrangement by doing the following

- Change the PivotTable form: Compact, Outline, or Tabular.
- Add, rearrange, and remove fields.
- Change the order of fields or items.

Change the layout of columns, rows, and subtotals by doing the following

- Turn column and row field headers on or off, or display or hide blank lines.
- Display subtotals above or below their rows.
- Adjust column widths on refresh.
- Move a column field to the row area or a row field to the column area.
- Merge or unmerge cells for outer row and column items.

Change the display of blanks and errors by doing the following

- Change how errors and empty cells are displayed.
- Change how items and labels without data are shown.
- Display or hide blank rows

Change the format by doing the following

- Manually and conditionally format cells and ranges.
- Change the overall PivotTable format style.
- Change the number format for fields.
- Include OLAP Server formatting.

Create Pivot table & Pivot chart

You can use data from a Excel worksheet as the basis for a PivotTable or PivotChart. The data should be in list format, with column labels in the first row, which Excel will use for Field Names. Each cell in subsequent rows should contain data appropriate to its column heading, and you shouldn't mix data types in the same column. For instance, you shouldn't mix currency values and dates in the same column. Additionally, there shouldn't be any blank rows or columns within the data range.

Excel tables are already in list format and are good candidates for PivotTable source data. When you refresh the PivotTable, new and updated data from the Excel table is automatically included in the refresh operation.

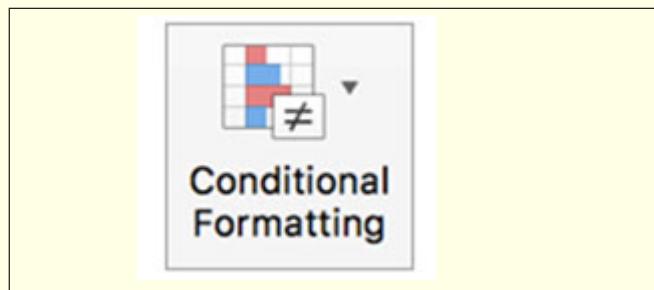
Using a dynamic named range: To make a PivotTable easier to update, you can create a dynamic named range, and use that name as the PivotTable's data source. If the named range expands to include more data, refreshing the PivotTable will include the new data.

Including totals: Excel automatically creates subtotals and grand totals in a PivotTable. If the source data contains automatic subtotals and grand totals that you created by using the Subtotals command in the Outline group on the Data tab, use that same command to remove the subtotals and grand totals before you create the PivotTable.

Conditional formatting

Apply conditional formatting

- 1 Select the range of cells, the table, or the whole sheet that you want to apply conditional formatting to.
- 2 On the Home tab, click Conditional Formatting.
- 3 Do one of the following:



To highlight	Do this
Values in specific cells. Examples are dates after this week, or numbers between 50 and 100, or the bottom 10% of scores.	Point to Highlight Cells Rules or Top/Bottom Rules, and then click the appropriate option.
The relationship of values in a cell range. Extends a band of color across the cell. Examples are comparisons of prices or populations in the largest cities.	Point to Data Bars, and then click the fill that you want.
The relationship of values in a cell range. Applies a color scale where the intensity of the cell's color reflects the value's placement toward the top or bottom of the range. An example is sales distributions across regions.	Point to Color Scales, and then click the scale that you want.
A cell range that contains three to five groups of values, where each group has its own threshold. For example, you might assign a set of three icons to highlight cells that reflect sales below \$80,000, below \$60,000, and below \$40,000. Or you might assign a 5-point rating system for automobiles and apply a set of five icons.	Point to Icon Sets, and then click a set.

Power Queries

With Power Query (known as Get & Transform in Excel), you can import or connect to external data, and then shape that data, for example remove a column, change a data type, or merge tables, in ways that meet your needs. Then, you can load your query into Excel to create charts and reports. Periodically, you can refresh the data to make it up to date. Power Query is available on three Excel applications, Excel for Windows, Excel for Mac and Excel for the Web. For a summary of all Power Query help topics, see Power Query for Excel Help.

The four phases of Power Query

There are four phases to using Power Query.

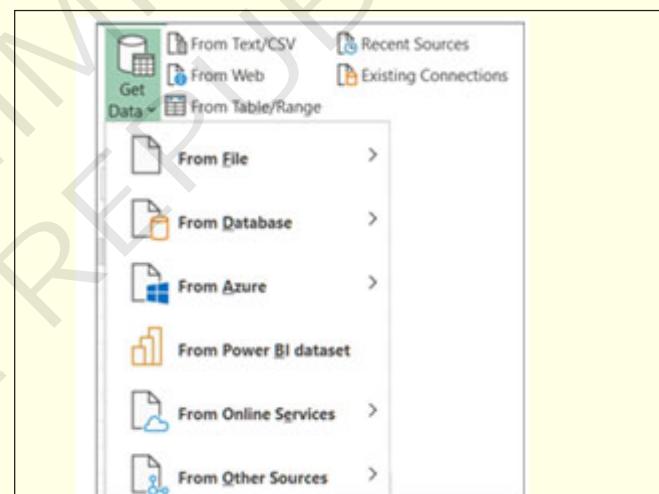


- Connect:** Make connections to data in the cloud, on a service, or locally
- Transform:** Shape data to meet your needs, while the original source remains unchanged
- Combine:** Integrate data from multiple sources to get a unique view into the data
- Load:** Complete your query and load it into a worksheet or Data Model and periodically refresh it.

The following sections explore each phase in more detail.

Connect: You can use Power Query to import to a single data source, such as an Excel workbook, or to multiple databases, feeds, or services scattered across the cloud. Data sources include data from the Web, files, databases, Azure, or even Excel tables in the current workbook. With Power Query, you can then bring all those data sources

together using your own unique transformations and combinations to uncover insights you otherwise wouldn't have seen.



Once imported, you can refresh the data to bring in additions, changes, and deletes from the external data source. For more information, see Refresh an external data connection in Excel.

Transform

Transforming data means modifying it in some way to meet your data analysis requirements. For example, you can remove a column, change a data type, or filter rows. Each of these operations is a data transformation. This process of applying transformations (and combining) to one or more sets of data is also called shaping data.

Think of it this way. A vase starts as a lump of clay that one shapes into something practical and beautiful. Data is the same. It needs shaping into a table that is suitable for your needs and that enables attractive reports and dashboards.

Power Query uses a dedicated window called the Power Query Editor to facilitate and display data transformations. You can open the Power Query Editor by selecting Launch

Query Editor from the Get Data command in the Get & Transform Data group, but it also opens when you connect to a data source, create a new query, or load a query.

The Power Query Editor keeps track of everything you do with the data by recording and labelling each transformation, or step, that you apply to the data. Whether the transformation is a data connection, a column removal, a merge, or a data type change, you can view and modify each transformation in the APPLIED STEPS section of the Query Settings pane.

There are many transformations you can make from the user interface. Each transformation is recorded as a step in the background. You can even modify and write your own steps using the Power Query M Language in the Advanced Editor.

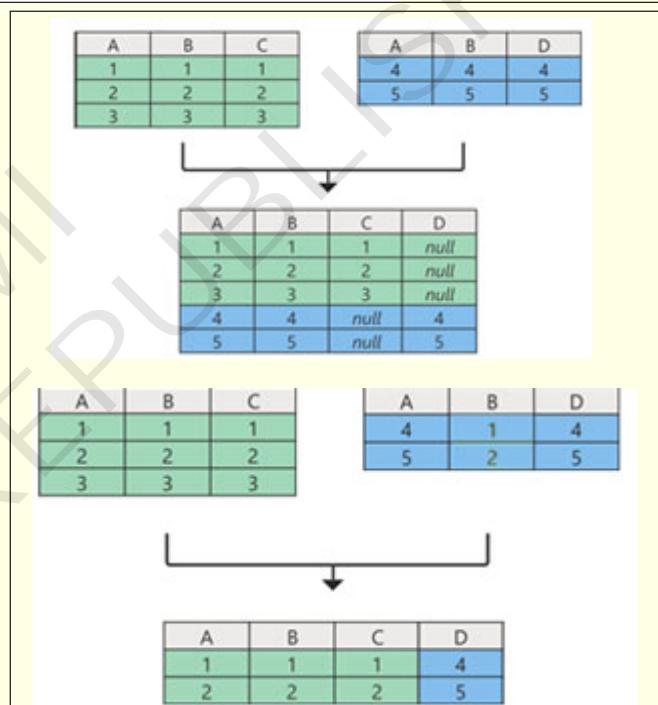
All the transformations you apply to your data connections collectively constitute a query, which is a new representation of the original (and unchanged) data source. When you refresh a query, each step runs automatically. Queries replace the need to manually connect and shape data in Excel.

Combine: You can combine multiple queries in your Excel workbook by appending or merging them. The Append and Merge operations are performed on any query with a tabular shape and are independent of the data sources that the data comes from.

Append : An append operation creates a new query that contains all rows from a first query followed by all rows from a second query. You can perform two types of append operations:

- **Intermediate Append:** Creates a new query for each append operation.
- **Inline Append:** Appends data to your existing query until you reach a final result.

Merge: A merge operation creates a new query from two existing queries. This one query contains all columns from a primary table, with one column serving as a navigation link to a related table. The related table contains all rows that match each row from a common column value in the primary table. Furthermore, you can expand or add columns from a related table into a primary table.



Load

There are two main ways to load queries into your workbook:

- From the Power Query Editor, you can use the Close and Load commands in the Close group on the Home tab.
- From the Excel Workbook Queries pane (Select Queries & Connections), you can right-click a query and select Load To.

You can also fine-tune your load options by using the Query Options dialog box (Select File > Options and settings > Query Options) to select how you want to view your data and where you want to load the data, either in a worksheet or a Data Model (which is a relational data source of multiple tables that reside in a workbook).

Application support for Power Query in Excel

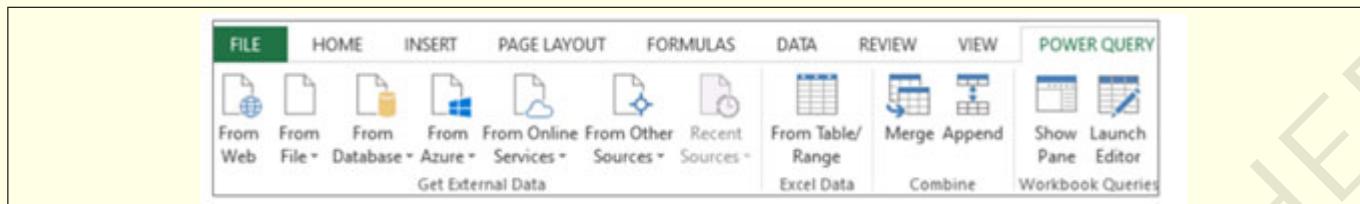
For over ten years, Power Query has been supported on Excel for Windows. Now, Excel is broadening Power Query support on Excel for Mac and adding support to Excel for the Web. This means we are making Power Query available on three major platforms and demonstrates the popularity and functionality of Power Query among Excel customers. Watch for future announcements on the Microsoft 365 roadmap and What's new in Excel for Microsoft 365.

A history of Power Query in Excel

The integration of Get & Transform Data (now called Power Query), into Excel has gone through a number of changes over the years.

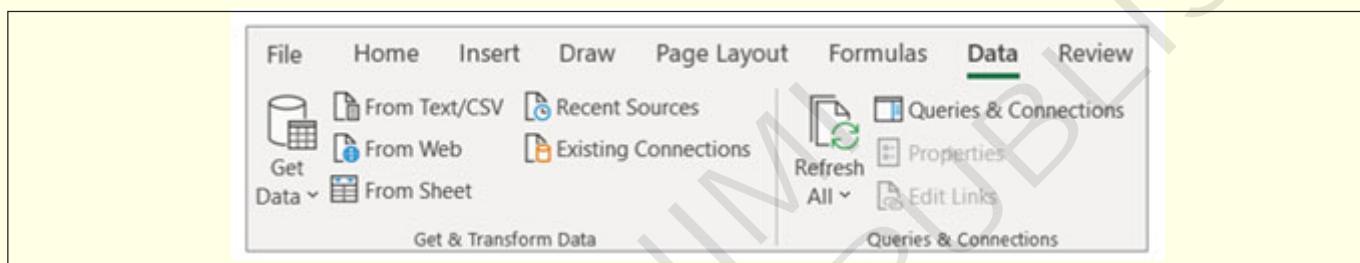
Excel 2010 and 2013 for Windows

In Excel 2010 for Windows, we first introduced Power Query and it was available as a free add-in that could be downloaded from here: Download the Power Query add-in. Once enabled, Power Query functionality was available from the Power Query tab on the ribbon.



Microsoft 365 : We updated Power Query to be the primary experience in Excel for importing and cleaning data. You can access the Power Query data import wizards and

tools from the Get & Transform Data group on the Data tab of the Excel ribbon.



This experience included enhanced data import functionality, rearranged commands on the Data tab, a new Queries & Connection side pane, and the continuing ability to shape data in powerful ways by sorting, changing data types, splitting columns, aggregating the data, and so on.

This new experience also replaced the older, legacy data import wizards under the Data command in the Get External Data group. However, they can still be accessed from the Excel Options dialog box (Select File > Options > Data > Show legacy data import wizards).

Excel 2016 and 2019 for Windows

We added the same Get & Transform Data experience based on the Power Query technology as that of Microsoft 365.

Excel for Microsoft 365 for Mac

In 2019 we started the journey to support Power Query in Excel for Mac. Since then, we added the ability to refresh Power Query queries from TXT, CSV, XLSX, JSON and XML files. We have also added the ability to refresh data from SQL server and from tables & ranges in the current workbook.

In October of 2019, we added the ability to refresh existing Power Query queries and to use VBA to create and edit new queries.

In January of 2021, we added support for the refresh of Power Query queries from OData and SharePoint sources.

For more information, see Use Power Query in Excel for Mac.

Note: There is no support for Power Query on Excel 2016 and Excel 2019 for Mac.

Data Catalog deprecation

With the Data Catalog, you could view your shared queries, and then select them to load, edit, or otherwise use in the current workbook. This feature was gradually deprecated:

- On August 1st, 2018, we stopped onboarding new customers to the Data Catalog.
- On December 3rd, 2018, users couldn't share new or updated queries in the Data Catalog.
- On March 4th, 2019, the Data Catalog stopped working. After this date, we recommended downloading your shared queries so you could continue using them outside the Data Catalog, by using the Open option from the My Data Catalog Queries task pane.

Power Query add-in deprecation

Early in the summer of 2019, we officially deprecated the Power Query add-in which is required for Excel 2010 and 2013 for Windows. As a courtesy, you may still use the add-in, but this may change at a later date.

E-commerce scope and benefits

Objectives: At the end of this lesson you shall be able to

- define E-commerce
- explain difference between traditional commerce and E-commerce
- explain type, scope and benefit of E-commerce.

E Commerce

E Commerce is the process of buying and selling products or service and transfer of funds electronically.

Electronic commerce draws on technologies such as mobile commerce, electronic funds transfer, supply chain management, Internet marketing, online transaction processing, electronic data interchange (EDI), inventory management systems, and automated data collection systems. Modern electronic commerce typically uses the World Wide Web for at least one part of the transaction's life cycle, although it may also use other technologies such as e-mail.

E-commerce businesses usually employ some or all of the following practices:

Provide Etail or "virtual storefronts" on websites with online catalogs, sometimes gathered into a "virtual mall".

Buy or sell on websites or online market places.

Gather and use demographic data through web contacts and social media.

Use electronic data interchange, the business-to-business exchange of data.

Reach prospective and established customers by e-mail or fax (for example, with newsletters).

Use business-to-business buying and selling.

Provide secure business transactions.

Flipkart's Big Billion Day sale on October 6, 2014 was touted as the mother of all online sales, but on the big day, things went awry and customers were left feeling high and dry. While the founders promptly apologised about the fact that they were not adequately prepared, the fact remains that ours is a nation that is latching on to e-commerce in a big way.

A single incident like this will not do much to dampen the interest in the \$ 3 billion sector that is poised for explosive growth. Already competitors of Flipkart such as Amazon and Snapdeal have learnt from this experience and probably try not to repeat such incidents in future.

As per The Associate Chambers of Commerce and Industry in India (ASSOCHAM), online shopping continues to rule the roost and will cross Rs 10,000 crore mark which is 350 per cent more than last year's volumes.

Traditional Commerce Vs. E Commerce:

In traditional commerce the buyer has to go to a shop to buy a product or service, but in E commerce, the product or services are listed in a web site and the buyer chooses the product by verifying its photos and other specifications given in the web sites and then paying the price preferably electronically by debit/credit card or Internet banking method. Then the product is delivered to the customer address by the company. Delivery charge may be free or it may be paid along with the payment of product cost.

In traditional Commerce the customer can physically verify the product for example if someone is buying an LED TV can go to a shop and personally verify the picture quality himself/herself, but in e commerce website it is not possible. Only buyer can see the photos of the product but cannot watch the picture quality or sound quality by himself/herself.

In e commerce, the customer can save his time and money by not going to the shop physically. But the customer needs an Internet connection and device to buy products.

In traditional commerce the buyer can physically meet the seller so if after sale service is required, the buyer can contact the seller but in e commerce the buyer does not get the chance to physically meet the seller. So after sale service is to be arranged by customer by contacting the service centers of respective product.

The product prices are also a factor in traditional commerce as products should be stored in shops and employees are also recruited to run the shop, the price goes high as there are rent for shop or it has to be owned the seller. In e commerce products are not transported to shop so the overhead of maintaining a shop is not added to the product cost. So products are cheaper in e commerce sites.

Though there are few people who can get benefit from e commerce sites. Mostly who stay in urban areas gets benefit. Because the service is normally restricted to urban areas only as villagers are not well equipped to use internet enabled devices and making online transactions. That's why e commerce sites first checks buyers pin code number to verify whether the product can be shipped to buyer or not.

With the advancement of technology in future this service can be availed by all over the country.

With the advent of e commerce sites like flipkart, amazon etc. traditional shop has seen a fall in sell as most urban buyer now preferring e commerce sites to buy products as they can buy product by sitting in the comfort of their home.

The customer can get the opportunity to see more products in e commerce sites than in shop. So the option is more and e commerce sites also supply the product to the buyers home which shops normally does not do.

Type of E Commerce and its Scope:

E-commerce can be divided into four categories, which are business to business b2b, business to customer b2c, government to business g2b and government to citizen g2c.

There are four main areas in which companies conduct E-commerce these areas are:

- Direct marketing, selling, and service.
- Online banking and billing.
- Secure distribution of information.
- Value chain trading and corporate purchasing.
- Filling tax return to government.

The field of E-Commerce is very broad. There are many applications of E-Commerce such as home banking, shopping in electronic malls, buying stocks, finding a job, conducting an auction, collaborating electronically with business partners around the globe, and providing customer service. The implementation of various E-Commerce applications depends on four major support categories such as people, public policy, and marketing/advertising and supply chain logistics. In addition there has to be an Infrastructure support. The E-Commerce management within each organization co-ordinates the applications and infrastructure. In order to explain the relationships I have explained below the applications in the case of B2C E-Commerce.

Benefits of E Commerce:

- E Commerce allows people to carry out businesses without the barriers of time or distance. One can log on to the Internet at any point of time, be it day or night and purchase or sell anything one desires at a single click of the mouse.
- The direct cost-of-sale for an order taken from a web site is lower than through traditional means (retail, paper based), as there is no human interaction during the on-line electronic purchase order process. Also, electronic selling virtually eliminates processing errors, as well as being faster and more convenient for the visitor.

- E Commerce is ideal for niche products. Customers for such products are usually few. But in the vast market place i.e. the Internet, even niche products could generate viable volumes.
- Another important benefit of E Commerce is that it is the cheapest means of doing business.
- The day-to-day pressures of the market place have played their part in reducing the opportunities for companies to invest in improving their competitive position. A mature market, increased competitions have all reduced the amount of money available to invest. If the selling price cannot be increased and the manufactured cost cannot be decreased then the difference can be in the way the business is carried out. Ecommerce has provided the solution by decreasing the costs, which are incurred.
- From the buyer's perspective also ecommerce offers a lot of tangible advantages.
 - 1 Reduction in buyer's sorting out time.
 - 2 Better buyer decisions.
 - 3 Less time is spent in resolving invoice and order discrepancies.
 - 4 Increased opportunities for buying alternative products.
- The strategic benefit of making a business 'E Commerce enabled', is that it helps reduce the delivery time, labour cost and the cost incurred in the following areas:
 - 1 Document preparation
 - 2 Error detection and correction
 - 3 Reconciliation
 - 4 Mail preparation
 - 5 Telephone calling
 - 6 Credit card machines
 - 7 Data entry
 - 8 Overtime
 - 9 Supervision expenses
- Operational benefits of e commerce include reducing both the time and personnel required to complete business processes, and reducing strain on other resources. It's because of all these advantages that one can harness the power of ecommerce and convert a business to E Business by using powerful E Commerce solutions made available by E Business solution providers.

Undertake transactions on an e-commerce site

Objectives: At the end of this lesson you shall be able to

- explain different E-commerce sites
 - explain online catalogues, shopping carts and check out pages
 - explain payment, order processing and authorization, charge back
 - explain other payment options.
-

Different E Commerce sites

Some of the world's most popular E Commerce sites are: Crate & Barrel, Symantec, Amway, Microsoft, Amazon, HP etc.

In India, after some initial hiccups, E Commerce is gradually picking pace. Some of the popular web sites are:

Amazon, FlipKart, Jabong, Naaptol etc.

On line catalogue, shopping carts and checkout pages

On line catalogues are list of products given on a web site for sell. Buyer chooses the product by browsing through the products and choosing the product which suits him/her.

Shopping cart is a bucket full of products chosen by buyer before finally paying the price. It is used in retail stores but the same concept has been implemented in Web site by making a virtual bucket which shows the product chosen by the buyer.

After choosing the desired product the buyer finally click checkout to pay the price for the products.

Payment and order processing

After clicking checkout a payment option is shows which normally has various options like COD (Cash on Delivery), Internet Banking, Debit Card, Credit Card and various other

options. The buyer chooses the proper option for payment and after successful payment, it is notified to the buyer and finally products are delivered to buyer address.

Authorization and chargeback

Authorization or authorization is the function of specifying access rights to resources related to information security and computer security in general and to access control in particular.

Chargeback refers to paying the money back to the buyer after the price has been deducted from his. It happens in various situations. For an example, suppose someone buy a ticket in irstc web site and paid the price of the ticket successfully, but later the ticket was not booked. Then charge back will occur and the money would be refunded back to the customer.

Other payment options

Apart from the above discussed payment option there are some other ways for payment exists like mobile payment. Recently Airtel Money or Vodafone mpesa etc. mobile payment methods has evolved so that, persons can pay by their mobile also.

Various E Commerce transactions like paying utility bills, shopping from web sites, recharging etc can be done by mobile.

E Commerce Security issues and Payment Gateways

Objectives: At the end of this lesson you shall be able to

- explain E-commerce security issue
- explain payment gateway.

E Commerce security issue

E-commerce security is the protection of e-commerce assets from unauthorized access, use, alteration, or destruction. While security features do not guarantee a secure system, they are necessary to build a secure system.

This massive increase in the uptake of eCommerce has led to a new generation of associated security threats, but any eCommerce system must meet four integral requirements:

- a Privacy - information exchanged must be kept from unauthorized parties.
- b Integrity - the exchanged information must not be altered or tampered with.
- c Authentication - both sender and recipient must prove their identities to each other and
- d Non-repudiation - proof is required that the exchanged information was indeed received.

Security Threats

Technical attacks are one of the most challenging types of security compromise an e-commerce provider must face. Perpetrators of technical attacks, and in particular Denial-of-Service attacks, typically target sites or services hosted on high-profile web servers such as banks, credit card payment gateways, large online retailers and popular social networking sites.

Denial of Service Attacks

Denial of Service (DoS) attacks consist of overwhelming a server, a network or a website in order to paralyze its normal activity. Defending against DoS attacks is one of the most challenging security problems on the Internet today. A major difficulty in thwarting these attacks is to trace the source of the attack, as they often use incorrect or spoofed IP source addresses to disguise the true origin of the attack.

The United States Computer Emergency Readiness Team defines symptoms of denial-of-service attacks to include:

- Unusually slow network performance
- Unavailability of a particular web site
- Inability to access any web site
- Dramatic increase in the number of spam emails received.

DoS attacks can be executed in a number of different ways including:

ICMP Flood (Smurf Attack)

Where perpetrators will send large numbers of IP packets with the source address faked to appear to be the address of the victim. The network's bandwidth is quickly used up, preventing legitimate packets from getting through to their destination.

Teardrop Attack

A Teardrop attack involves sending mangled IP fragments with overlapping, over-sized, payloads to the target machine. A bug in the TCP/IP fragmentation re-assembly code of various operating systems causes the fragments to be improperly handled, crashing them as a result of this.

Phlashing

Also known as a Permanent denial-of-service (PDoS) is an attack that damages a system so badly that it requires replacement or reinstallation of hardware. Perpetrators exploit security flaws in the remote management interfaces of the victim's hardware, be it routers, printers, or other networking hardware. These flaws leave the door open for an attacker to remotely 'update' the device firmware to a modified, corrupt or defective firmware image, therefore bricking the device and making it permanently unusable for its original purpose.

Distributed Denial-of-Service Attacks

Distributed Denial of Service (DDoS) attacks are one of the greatest security fears for IT managers. In a matter of minutes, thousands of vulnerable computers can flood the victim website by choking legitimate traffic. A distributed denial of service attack (DDoS) occurs when multiple compromised systems flood the bandwidth or resources of a targeted system, usually one or more web servers. The most famous DDoS attacks occurred in February 2000 where websites including Yahoo, Buy.com, eBay, Amazon and CNN were attacked and left unreachable for several hours each.

Brute Force Attacks

A brute force attack is a method of defeating a cryptographic scheme by trying a large number of possibilities; for example, a large number of the possible keys in a key space in order to decrypt a message. Brute Force Attacks, although perceived to be low-tech in nature are not a thing of the past.

Non-Technical Attacks

Phishing is the criminally fraudulent process of attempting to acquire sensitive information such as usernames, passwords and credit card details, by masquerading as a trustworthy entity in an electronic communication. Phishing scams generally are carried out by emailing the victim with a 'fraudulent' email from what purports to be a legitimate organization requesting sensitive information. When the victim follows the link embedded within the email they are brought to an elaborate and sophisticated duplicate of the legitimate organizations website. Phishing attacks generally target bank buyers, online auction sites (such as eBay), online retailers (such as amazon) and services providers (such as PayPal). According to community banker, in more recent times cyber criminals have got more sophisticated in the timing of their attacks with them posing as charities in times of natural disaster.

Social Engineering-Social engineering is the art of manipulating people into performing actions or divulging confidential information. Social engineering techniques include pretexting (where the fraudster creates an invented scenario to get the victim to divulge information), Interactive voice recording (IVR) or phone phishing (where the fraudster gets the victim to divulge sensitive information over the phone) and baiting with Trojans horses (where the fraudster 'baits' the victim to load malware unto a system). Social engineering has become a serious threat to e-commerce security since it is difficult to detect and to combat as it involves 'human' factors which cannot be patched akin to hardware or software, albeit staff training and education can somewhat thwart the attack.

How to be secure

Shop at Secure Web Sites

Secure sites use encryption technology to transfer information from your computer to the online merchant's computer. Encryption scrambles the information you send, such as your credit card number, in order to prevent computer hackers from obtaining it en route. The only people who can unscramble the code are those with legitimate access privileges. Here's how you can tell when you are dealing with a secure site:

If you look at the top of your screen where the Web site address is displayed (the "address bar"), you should see https://. The "s" that is displayed after "http" indicates that Web site is secure. Often, you do not see the "s" until you actually move to the order page on the Web site.

Another way to determine if a Web site is secure is to look for a closed padlock displayed on the address bar of your screen.

If that lock is open, you should assume it is not a secure site. Of course, transmitting your data over secure channels is of little value to you if the merchant stores the data unscrambled. You should try to find out if the merchant stores the data in encrypted form. If a hacker is able to intrude, it cannot obtain your credit data and other personal information. Be sure to read the merchant's privacy and security policies to learn how it safeguards your personal data on its computers.

Research the Web Site before You Order

Do business with companies you already know. If the company is unfamiliar, do your homework before buying their products. If you decide to buy something from an unknown company, start out with an inexpensive order to learn if the company is trustworthy.

Reliable companies should advertise their physical business address and at least one phone number, either buyer service or an order line. Call the phone number and ask questions to determine if the business is legitimate. Even if you call after hours, many companies have a "live" answering service, especially if they don't want to miss orders. Ask how the merchant handles returned merchandise and complaints. Find out if it offers full refunds or only store credits.

You can also research a company through the Better Business Bureau, or a government consumer protection agency like the district attorney's office or the Attorney General. Perhaps friends or family members who live in the city listed can verify the validity of the company. Remember, anyone can create a Web site.

Payment gateways

A payment gateway is an e-commerce application service provider service that authorizes credit card payments for e-businesses, online retailers, bricks and clicks, or traditional brick and mortar.

It is the equivalent of a physical point of sale terminal located in most retail outlets. Payment gateways protect credit card details by encrypting sensitive information, such as credit card numbers, to ensure that information is passed securely between the buyer and the merchant and also between merchant and the payment processor.

A payment gateway facilitates the transfer of information between a payment portal (such as a website, mobile phone or interactive voice response service) and the Front End Processor or acquiring bank.

Transaction process

- When a buyer orders a product from a payment gateway-enabled merchant, the payment gateway performs a variety of tasks to process the transaction.

Overview of information security and threats

Objectives: At the end of this lesson you shall be able to

- describe information security and its basic principles
- describe the relation between information security and cybersecurity
- describe the key challenges in information security
- describe the benefits of information security
- explain the methods of implementing information security.

Introduction

Information Security is the protection of information and information systems from unauthorized access, use, disclosure, disruption, modification, or destruction in order to provide confidentiality, integrity, and availability. It is a general term that can be used regardless of the form the data may take (electronic, physical, etc.).

Governments, military, corporations, financial institutions, hospitals and private businesses store a great deal of confidential information about their employees, customers, products, research and financial status. Most of this information is now collected, processed and stored on electronic computers and transmitted across networks to other computers.

If confidential information about a business' customers or finances or new product line falls into the hands of a competitor or a hacker, a business and its customers could suffer widespread, irreparable financial loss, not to mention damage to the company's reputation. Protecting confidential information is a business requirement and in many cases also an ethical and legal requirement.

Information assurance: Information assurance is the act of ensuring that data is not lost when critical issues arise. These issues include but are not limited to: natural disasters, computer/server malfunction, physical theft, or any other instance where data has the potential of being lost. Since most information is stored on computers in our modern era, information assurance is typically dealt with by IT security specialists. One of the most common methods of providing information assurance is to have an off-site backup of the data in case one of the mentioned issues arise.

Basic principles of Information Security: The CIA Triad is a well-known model for security policy development, used to identify problem areas and necessary solutions for information security. Confidentiality, integrity, and availability (CIA) is a model designed to guide policies for information security within an organization. The CIA triad of confidentiality, integrity, and availability is at the heart of information security. The members of the Classic Information Security triad -confidentiality, integrity and availability - are also referred to as security attributes, properties, security goals, fundamental aspects, information criteria, critical information characteristics and basic building blocks.

Confidentiality

Confidentiality is a set of rules that limits access to information. Confidentiality prevents sensitive information from reaching the wrong people, while making sure that the right people can in fact get it. Protecting confidentiality depends upon defining and enforcing appropriate access levels for information. This may be done by separating information into separate units organized by who should have access to it and how sensitive it is.

Integrity

Integrity is the assurance that the information is trustworthy, consistent and accurate over its entire life-cycle. This means that data cannot be modified in an unauthorized or undetected manner. Data must not be changed in transit, and steps must be taken to ensure that data cannot be altered by unauthorized people (for example, in a breach of confidentiality). In addition, some means must be in place to detect any changes in data that might occur as a result of non-human-caused events such as an electromagnetic pulse (EMP) or server crash. If an unexpected change occurs, a backup copy must be available to restore the affected data to its correct state.

Availability

Availability is a guarantee of ready access to the information by authorized people. For any information system to serve its purpose, the information must be available when it is needed. This means that the computing systems used to store and process the information, the security controls used to protect it, and the communication channels used to access it must be functioning correctly.

High availability systems aim to remain available at all times, preventing service disruptions due to power outages, hardware failures, and system upgrades. Availability is best ensured by rigorously maintaining all hardware, performing hardware repairs immediately when needed, providing a certain measure of redundancy and failover, providing adequate communications bandwidth and preventing the occurrence of bottle necks, implementing emergency backup power systems, keeping current with all necessary system upgrades, and guarding against malicious actions such as denial-of-service (DoS) attacks.

In addition to the above mentioned three members, Authenticity and Non-repudiation are also considered to be members of the CIA model.

Authenticity

Authenticity is the process of ensuring that the data, transactions, communications or documents are genuine. It is also important for authenticity to validate that the parties involved are genuine. Some information security systems incorporate authentication features such as "digital signatures", which give evidence that the messaged data is genuine and was sent by someone possessing the proper signing key.

Non-repudiation

Non-repudiation means a person's intention to fulfill his obligations to a contract. It also implies that one party of a transaction cannot say that they have not received a transaction nor can the other party deny having sent a transaction.

It is important to note that while technology such as cryptographic systems can assist in non-repudiation efforts, the concept is basically a legal concept. It is not, for instance, sufficient to show that the message matches a digital signature signed with the sender's private key, and thus only the sender could have sent the message and nobody else could have altered it in transit. The alleged sender could in return demonstrate that the digital signature algorithm is vulnerable or flawed, or allege or prove that his signing key has been compromised. The fault for these violations may or may not lie with the sender himself, and such assertions may or may not relieve the sender of liability, but the assertion would invalidate the claim that the signature necessarily proves authenticity and integrity and thus prevents repudiation.

With all activities that give us almost unlimited freedom, there are risks. Because the Internet is so easily accessible to anyone, it can be a dangerous place. Know who you're dealing with or what you're getting into. Predators, cyber criminals, bullies, and corrupt businesses will try to take advantage of the unwary visitor.

The internet, Intranet and Security

Although the difference between Intranets and the Internet is not great in terms of technology, the transmission of information is completely different from the organizational point of view.

Information security threats between Intranets and other networks and information systems are rather similar. The use of Intranets as internal information channels emphasizes the importance of information security. Assets held on internal Intranets may increase the interest of potential misusers. Hence, protecting Intranets and the data and information transmitted via them against various threats endangering the confidentiality, integrity and availability of information is an extremely important consideration.

Using the Internet as a part of an Intranet poses a serious threat, because the Internet is inherently nonsecure. As a result, users must be very careful particularly in encrypting their communications. Imitation (spoofing), reply (rapid fire), alteration of message contents (superzapping), prevention of service availability and active and passive wiretapping

are among the most malicious threats. Wiretapping, for example, could lead to a situation where strategic knowledge regarding an organization gets in the hands of outsiders, if communication encryption is not implemented by means of strong encryption methods.

Hacker tools, although developed for the Internet, are also usable on Intranets. They can be software or hardware based or a combination of both. Their authorized use includes finding and correcting information security weaknesses on Intranets. However, they also enable insiders to hack such communication systems and access information which they are not authorized to access.

Relation Between information Security and Cyber security

Information Security, mentioned in the earlier sections is the protection of information and information systems from unauthorized access, use, disclosure, disruption, modification, or destruction in order to provide confidentiality, integrity, and availability.

Cybersecurity on the other hand can be defined as the ability to protect or defend the use of cyberspace from cyber-attacks. Cyberspace is "the environment in which communication over computer networks occurs."

Cyber security involves anything security-related in the cyber domain or realm (or cyberspace). Information security involves the security of information or information systems regardless of the realm it occurs in (e.g., risk of exposure in physical world). Since anything that occurs in the cyber realm would involve the protection of information and information systems in some way, you can conclude that information security is a super-set of cyber security. (Fig 1) At times the two terms are used interchangeably too.

Key Challenges in Information Security

1 IT Security is assigned a low priority

- The organization have not instilled the right focus on implementing IT security practices.

2 Ad hoc Security Governance

- Absence of an Information Security Management System (ISMS) or a structured governance mechanism. (Fig 1)

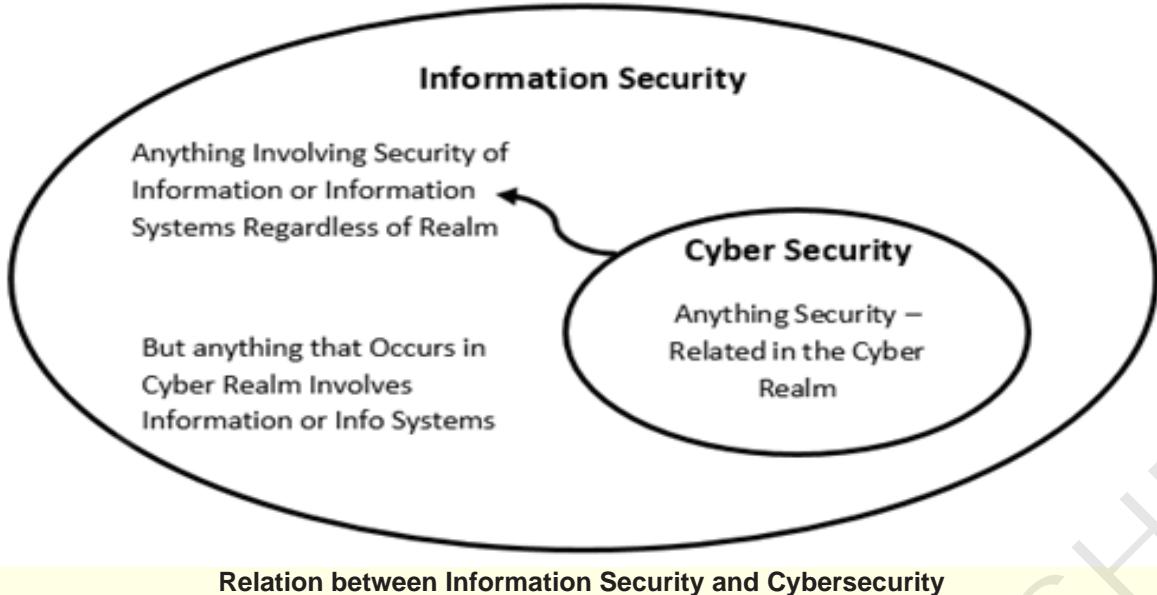
3 Ambiguity in roles and responsibilities

- Ambiguities exist on the roles and responsibilities of the different players (Business, teams in SSO, etc.) in an SSO. Single sign-on (SSO) is a property of access control of multiple related, but independent software systems. With this property a user logs in once and gains access to all systems without being prompted to log in again at each of them

4 Inadequate Separation of Duties

- Overlapping and shared responsibilities in an SSO makes it difficult to implement appropriate level of separation in duties.

Fig 1



5 Varied Interpretations of Security Requirements

- In the absence of standard interpretations, the different individuals and teams have their own interpretations.

6 Tendency to reduce Risk level

- The teams show a tendency to reduce the 'Risk Level' to bypass the rigors of the governing processes.

7 Multiple vendors

- Relentless competition and sense of insecurity have led to reluctance in sharing responsibility and little or no collaboration among the vendors.

8 Business/Operations spread across multiple geographics

- The organization is based out of and functions from multiple locations spread all across the globe.

9 Lack of Training/Awareness

- Inadequate training and awareness on security practices.

Benefits of Information Security:

- Protect networks, computers and data from unauthorized access to minimize the impact from external threats of various cybercrime
- Improved information security and business continuity management to implement technical, management, administrative and operational controls, which is the most cost effective way of reducing risk.
- Improved stakeholder confidence in information security arrangements.
- Improved company credentials with the correct security controls in place. Organization will improve credibility and trust among internal stakeholders and external vendors. The credibility and trust are the key factors to win a business.

- Faster recovery times in the event of disruption

Techniques to enforce IS in an organization

Identifying tools to enforce Information Security

A successful information security policy provides several benefits to corporations. Enforceable policies ensure that vulnerabilities are identified and addressed. This results in protecting business continuity and strengthening the IT infrastructure. When employees throughout an organization follow a security policy, ensuring that information is safely shared within the organization as well as with customers, partners and vendors, the risk is reduced.

1 The first step to creating an effective information security policy is evaluating information assets and identifying threats to those assets.

Some assets within an organization will be more valuable than others, but monetary value should not be the only factor. Determining both the monetary value and the intrinsic value of an asset is essential in accurately gauging its worth. To calculate an asset's monetary value, an organization should consider the impact if that asset's data, networks or systems are compromised in any way. To calculate intrinsic value, an organization must consider a security incident's impact on credibility, reputation and relationships with key stakeholders.

2 Creating a policy is for organizations to perform a risk assessment

After the identification of assets and threats, the next step in creating a policy is for organizations to perform a risk assessment. This assessment allows an organization to decide whether information is under protected, overprotected or adequately protected. The goal for this risk assessment should be to minimize expenses without exposing an organization to unnecessary risk. This assessment will help in determining the proper allocation of resources once the security policy is effectively in place.

The Information Security Framework

The Information Security Framework establishes security policy and practices for an organization. Policies provide guidance on matters affecting security that an organization's members are expected to follow. Security policy applies to all hardware, software, data, information, network, personal computing devices, support personnel, and users within an organization.

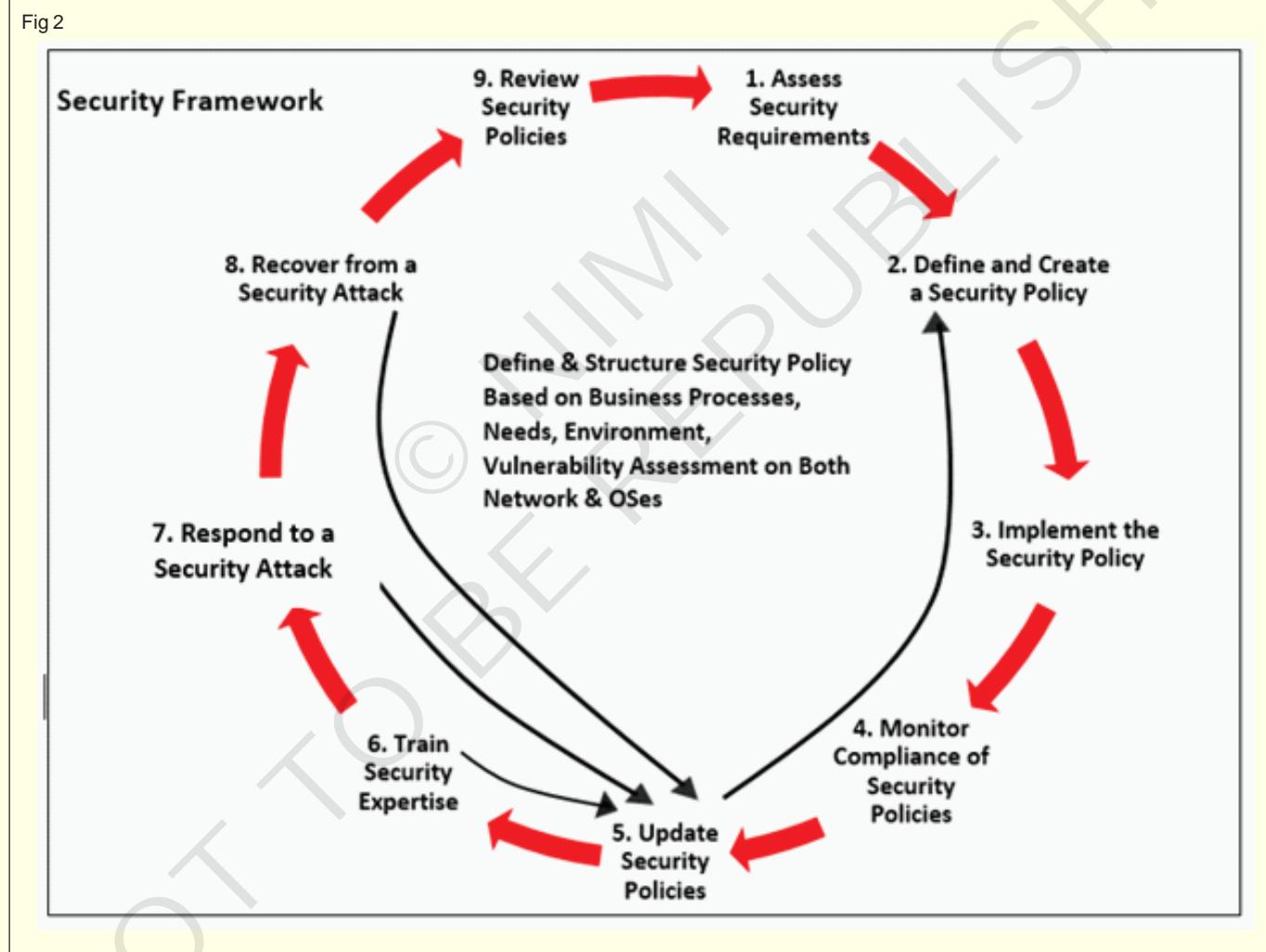
For an IT security system to work, a well-defined framework needs to be developed involving all stakeholders, and it needs to be updated over time to be useful. The information security frameworks facilitate the management process in considering the handling of data and implementation of system/ process in the form of identifying assets, determining security requirement, risk assessment, control evaluation, control implementation, process monitoring and update.

The commonly used terms in Information Security Framework are:

- Policy: General Management Statements
- Standards: Specific Mandatory Controls
- Guidelines: Recommendations / Best Practices
- Procedures: Step by Step Instructions

The detailed activities involved in actually implementing a Security Framework, the sequence of practice to be followed, corrective actions to be taken are shown in the Figure shown below. (Fig 2).

Fig 2



The major heads and the practices under each head in a Framework are as shown below.

Thus the security framework becomes the technology that turns security policies into practice. It achieves it by the four steps cycle of plan, do act and check cycle. The PPT triad, ie. people, process and technology needs to be given equal importance in achieving this.

New technologies and new networks can plug into the security framework and Security costs become more predictable and manageable.

Identify	Protect	Detect	Respond	Recover
<ul style="list-style-type: none"> • Asset Management • Business Environment • Governance • Risk • Risk Management Strategy 	<ul style="list-style-type: none"> • Access Control Events • Awareness and Training • Data Security • Information Protection and Procedures • Maintenance Assessment • Protective technology 	<ul style="list-style-type: none"> • Anomalies and Security Continuous Monitoring • Detection Process 	<ul style="list-style-type: none"> • Response Planning • Communications • Analysis • Mitigation • Improvements 	<ul style="list-style-type: none"> • Recovery Planning • Improvement • Communications

Overview of security threats

Objectives: At the end of this lesson you shall be able to

- describe security threat and its types
- describe the methods of identifying threats
- explain how threats affect a system
- describe the sources of threats
- describe the best practices to identify and mitigate threats.

Introduction

A Threat is any circumstance or event with the potential to cause harm to the system or activity in the form of destruction, disclosure, and modification of data, or denial of service. A threat is a potential for harm. The presence of a threat does not mean that it will necessarily cause actual harm.

Some of the common terms associated with threats and their description are as follows:

1 Unauthorized Access

The attempted or successful access of information or systems, without permission or rights to do so.

2 Cyber Espionage

The act of spying through the use of computers, involving the covert access or 'hacking' of company or government networks to obtain sensitive information.

3 Malware

A collective term for malicious software, such as viruses, worms and trojans; designed to infiltrate systems and information for criminal, commercial or destructive purposes.

4 Data Leakage

The intentional or accidental loss, theft or exposure of sensitive company or personal information.

5 Mobile Device Attack

The malicious attack on, or unauthorized access of, mobile devices and the information stored or processed by them; performed wirelessly or through physical possession.

6 Social Engineering

Tricking and manipulating others by phone, email, online or in-person, into divulging sensitive information, in order to access company information or systems.

7 Insiders

An employee or worker with malicious intent to steal sensitive company information, commit fraud or cause damage to company systems or information.

8 Phishing

A form of social engineering, involving the sending of legitimate looking emails aimed at fraudulently extracting sensitive information from recipients, usually to gain access to systems or for identity theft.

9 System Compromise

A system that has been attacked and taken over by malicious individuals or 'hackers', usually through the exploitation of one or more vulnerabilities, and then often used for attacking other systems.

9 Spam

Unsolicited email sent in bulk to many individuals, usually for commercial gain, but increasingly for spreading malware.

10 Denial of Service

An intentional or unintentional attack on a system and the information stored on it, rendering the system unavailable and inaccessible to authorized users.

11 Identity Theft

The theft of an unknowing individual's personal information, in order to fraudulently assume that individual's identity to commit a crime, usually for financial gain.

Categories of threats

Security Threats can be classified in many ways. A few of the popular classifications are as follows:

1 Based on the sophistication, Security threats can be classified into three categories.

- Simple first-generation threats are generic virus-type attacks spread by users opening infected e-mail and inconspicuous file attachments.
- The second-generation threats are more sophisticated and pose bigger problems. Created with automated tools, these worms attack vulnerabilities without human interaction. Replication, identification, and targeting of new victims is automatic.
- The third generation threats are blended threats, are common and incorporate viruses, Trojans and automation. These worms pre-compile targets for hyper-propagation, exploit known vulnerabilities and enable targeted use of hidden vulnerabilities. They also target multiple attack wireless links, virtual private networks and attack inside perimeter defences such as firewalls and intrusion detection systems.

2 The top threats according to OWASP (Open Web Applications Security Project) are as follows:

- Injection
- Cross Site Scripting (CSS)
- Broken Authentication and Session Management
- Insecure Direct Object References
- Cross - Site Request Forgery (CSRF)
- Security Misconfiguration
- Insecure Cryptographic Storage
- Failure to Restrict URL Access
- Insufficient Transport Layer Protection
- Unvalidated Redirects and Forwards

3 Categorization by Microsoft according to the kinds of Exploits that are used (or motivation of the attacker) ie. STRIDE system. The STRIDE acronym is formed from the first letter of each of the following categories.

- Spoofing identity. An example of identity spoofing is illegally accessing and then using another user's authentication information, such as username and password.
- Tampering with data. Data tampering involves the malicious modification of data.
- Repudiation. Repudiation threats are associated with users who deny performing an action without other parties having any way to prove otherwise-for example, a user performs an illegal operation in a system that lacks the ability to trace the prohibited operations.

- Information disclosure. Information disclosure threats involve the exposure of information to individuals who are not supposed to have access to it.
- Denial of service. Denial of service (DoS) attacks deny service to valid users.
- Elevation of privilege. In this type of threat, an unprivileged user gains privileged access and thereby has sufficient access to compromise or destroy the entire system.

Threats based on technology

- 1 Threats based on WWW technology
- 2 New features in browser software
- 3 Browser software test versions
- 4 Server software
- 5 CGI scripts
- 6 Cookies
- 7 Threats based on Unix and TCP/IP tools
- 8 Difficulties in firewall management
- 9 Use of cryptographic software
- 10 Hacker tools
- 11 Other software based threats
- 12 Intranet application software
- 13 Java language
- 14 ActiveX
- 15 Threats based on communications
- 16 Threats based on viruses
- 17 Threats based on human activities

Identification of Information Security Threats

The success of an information security management program is based on the accurate identification of the threats to the organization's information systems. Identification of Information Security Threats is an essential first step for security planners. Proper threat and vulnerability identification should include security testing and inspections, which are geared to promoting and ensuring that equipment is operating properly, is readily available when needed, and that employees are proficient in the use of the equipment.

To accomplish this, systems must design a testing program that not only assesses the current state of security, but can also be used to upgrade staff effectiveness through training.

The two major methods of identifying security threats are **Probing** and **Scanning**.

Probing is an attempt to gain access to a computer and its files through a known or probable weak point in the computer system. It is an action taken for the purpose of learning something about the state of the network.

Scanning is a method to go through all the files, or network elements with an intention to detect something unusual. File scanning inspects files that users attempt to download or open remotely for viruses and other malicious content. File scanning returns some information for policy enforcement.

There are 2 types of file scanning. They can be used together.

- Advanced detection applies techniques to discover known and emerging threats, including viruses, Trojan horses, worms, and others.
- Anti-virus scanning uses anti-virus definition files to identify virus-infected files.

Network scanning is a procedure for identifying active hosts on a network for the purpose of network security assessment. Scanning procedures, such as ping sweeps and port scans, return information about which IP addresses map to live hosts that are active on the Internet and any suspicious activity etc.

Modus Operandi of Threats in attacking a system

Typically an attack on a device takes place in 3 phases:

- The infection of a host,
- The accomplishment of its goal, and
- The spread of the malware to other systems.

Threats often use the resources offered by the infected devices. They use the output devices such as Bluetooth or infrared, but may also use the address book or email address of the person to infect the user's acquaintances.

They exploit the trust that is given to data sent by an acquaintance.

Infection

Infection is the means used by the threat to get into the device. It can either use one of the faults previously presented or may use the gullibility of the user. Infections may ask for permission, or may interact with the gullible user or may not even do any of these two and directly attack the system.

Accomplishment of the goal

Once the threat has infected a device it will also seek to accomplish its goal, which is usually one of the following: hardware damage, denial of service(DoS), monetary damage, damage data and/or , device, and concealed damage etc.

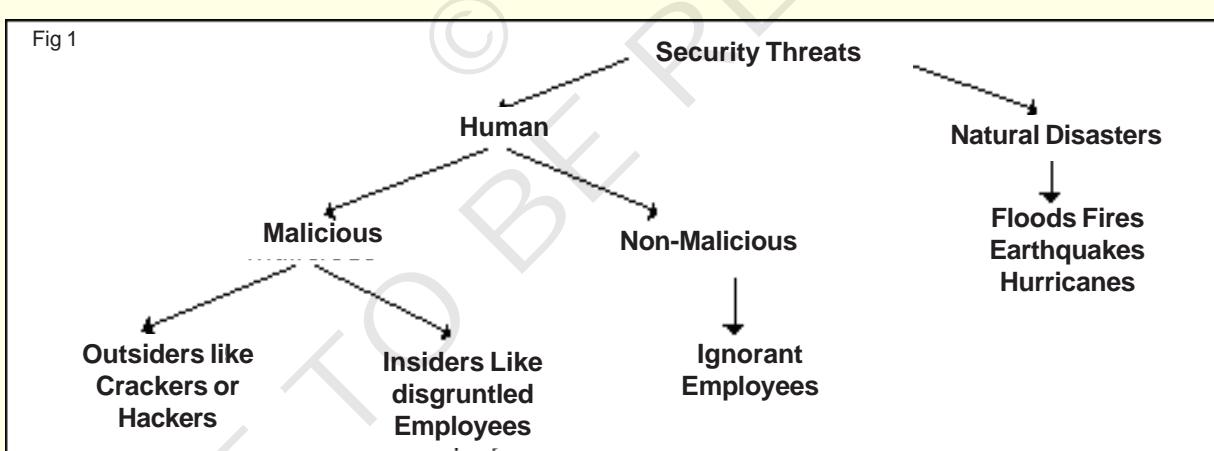
Spreading to other systems

Once the threat has infected a device, it always aims to spread one way or another.

It can spread through networks, wired or wireless, through the internet, proximate devices using Wi-Fi, Bluetooth and infrared light and through shared devices etc.

Sources of Threats (Fig 1)

Primary sources of threats are employees/insiders, malicious hackers, natural disasters, foreign adversaries, and hostile attacks. In several cases, the areas for sources of threats may overlap. For example, hostile attacks may be performed by foreign adversaries or a disgruntled employee.



Natural Disasters: Earthquakes, hurricanes, floods, lightning, and fire can cause severe damage to computer systems. Information can be lost, down time or loss of productivity can occur, and damage to hardware can disrupt other essential services. Few safeguards can be implemented against natural disasters. The best approach is to have disaster recovery plans and contingency plans in place.

Human Threats

A Employees/Insiders: Intentional and accidental errors and malicious acts by employees and insiders cause a considerable amount of damages and losses experienced in the telecommunications industry.

Disgruntled employees can create both mischief and sabotage on a computer system. Staff removed from their jobs in both public and private sectors has created a group of individuals with important organizational knowledge who may retain potential system access. System managers can limit this threat by invalidating passwords and deleting system accounts in a timely manner.

Malicious Hackers

Malicious threats consist of inside attacks by disgruntled or malicious employees and outside attacks by non-employees just looking to harm and disrupt an organization. Malicious attackers normally will have a specific goal, objective, or motive for an attack on a system.

These goals could be to disrupt services and the continuity of business operations by using denial-of-service (DoS) attack tools. They might also want to steal information or even steal hardware such as laptop computers. Hackers can sell information that can be useful to competitors.

The most dangerous attackers are usually insiders (or former insiders), because they know many of the codes and security measures that are already in place. Insiders are likely to have specific goals and objectives, and have legitimate access to the system. Employees are the people most familiar with the organization's computers and applications, and they are most likely to know what actions might cause the most damage. Insiders can plant viruses, Trojan horses, or worms, and they can browse through the file system.

However, disgruntled current employees actually cause more damage than former employees. Common examples of computer-related employee sabotage include:

- Changing data
- Deleting data
- Destroying data or programs with logic bombs
- Crashing systems
- Holding data hostage
- Destroying hardware facilities
- Entering data incorrectly

Foreign Adversaries

Computer intruder activities have occurred internationally, with the number of attempted intrusions through international gateways is increasing at an alarming rate. There have been few indications that computer under grounds within foreign countries carry over political agendas. Sometimes intelligence services of one country directly target, penetrate, or compromise the Communications systems of other countries.

Outside Attackers or 'Crackers'

People often refer to "crackers" as "hackers." The term hacker refers to people who either break in to systems for which they have no authorization or intentionally overstep their bounds on systems for which they do not have legitimate access. Common methods for gaining access to a system include password cracking, exploiting known security weaknesses, network spoofing, and social engineering.

Hostile Attacks

Through hostile attacks, it is possible to affect the availability of the networks. The primary impact of hostile attacks such as coordinated nuclear attacks, limited/uncoordinated nuclear attacks, nuclear accidents, terrorism, electronic warfare, sabotage, and civil disorder on the a nation's network is disruption and denial of service. Such disasters impact the timeliness and quality of the delivered services.

B Non-Malicious Employees

Attackers are not the only ones who can harm an organization. The primary threat to data integrity comes from ignorant users. These are authorized users who are not aware of the actions they are performing. Errors and omissions can lose, damage, or alter valuable data.

Users, data entry clerks, system operators, and programmers frequently make unintentional errors that contribute to security problems, directly and indirectly. Sometimes the error is the threat, such as a data entry error or a programming error that crashes a system. In other cases, errors create vulnerabilities.

Best Practices or Guidelines used to Identify Threats

Different mechanisms and methodologies can be used to successfully identify threats/attacks depending on their type. In other words, depending on the threat, you can use specific techniques to identify and classify them accordingly. Following are the most common methodologies:

- The use of anomaly detection tools
- Network telemetry using flow-based analysis
- The use of intrusion detection and intrusion prevention systems (IDS/IPS)
- Analyzing network component logs (that is, SYSLOG from different network devices, accounting records, application logs, Simple Network Management Protocol (SNMP) etc.

Best Practices or Guidelines used in mitigation of threats

The following are some of the practices which when implemented are likely to reduce the threats to the security of an organization's information, data and credibility.

Security Awareness Training

- Most security breaches actually originate inside companies by disgruntled or negligent employees. So educate everyone in your company so they can help identify a variety of security risks.
- Employees should be able to spot and identify email phishing and spoofing attacks. They should also be trained not to store, send or copy sensitive information that's unencrypted.
- They should know not to share sensitive information over the phone unless they are 100% sure of the audience.
- Train employees on security policies and practices. And make sure to update them and retrain at frequently.
- Training materials should also review corporate policies and clearly detail consequences for any suspicious or malicious behavior amongst employees. They should be trained on various security policies, including:
 - Acceptable Use and Unacceptable Use
 - General Use and Ownership

- Security & Proprietary Information
- Blogging & Social Media
- Enforcement ie. the disciplinary action for non-compliance
- Social Media
- Bring Your Own Device Policies
 - Data Policy
 - Mobile Device Management (MDM) Policy
 - Mobile Device Support Policy
 - Policies Regarding Company-issued Devices
 - Loss & Theft
 - Employee Termination Policy
 - Security Incident Management

Anti-Virus & Anti-Malware Protection

- Common forms of malware include: Worms, Key loggers, Video frame grabbers, Rootkits and Trojan horses.
- Install, update, schedule and run good Antivirus programs.
- Adopt an "end point security" strategy to combat malware threats. Endpoint security is an information security concept that means that each device (or endpoint) on a network should be responsible for and capable of providing for its own security.
- Whatever your anti-malware solution, it should scan email for attached viruses, monitor files in real time for infections, and perform thorough scans of every file.

Data Encryption

- Data encryption is a powerful part of information security. Encryption protects your data even after it has been accessed.

Patching

- Patching is essential to minimizing the risk to your computer systems. Patches are often released to fix security holes in systems and applications. Make sure you keep all operating systems and applications you run patched. Install the latest firmware updates on all network devices.

Access Controls

- For increased security, give employees only (and partners) access to the data they need. This includes both physical and logical access.
- Start by granting the least privilege. You can then escalate privileges to allow access to unauthorized data on an as-needed basis.

Mobile Devices

- Laptops, smartphones, and tablets have increased the productivity and mobility of today's workforce. But along with that productivity comes vulnerability. Lost or stolen laptops and other mobile devices are the top cause of data breaches.
- Enable auto-lock or require a password to access all devices.

Monitoring

- Make sure your business is set up to monitor systems and network devices for any abnormalities.
- Collect and correlate information from all places or infrastructure - network, systems, and user activity.
- Don't just block activity at a firewall or IPS. Log it, review it and learn from it.
- Install content filtering to monitor user activity from within your business. The most common form of employee misuse of the Internet is to surf unwanted sites

Firewall

- Configure Firewall rules and Policies because a firewall is the first line of defense against any attack (network or host). It acts a barrier between a public network and a private network.

Remote Backup

- Backup your data regularly to a remote location. Backup is one of the most neglected areas of computing and therefore typically one of the biggest opportunities your business has to mitigate risk.
- Often, businesses invest in securing data from hackers or malware, but then the data is physically destroyed by natural causes. If the data doesn't exist, securing it from outside threats has no meaning.

Security Assessments & Penetration Testing

- To secure your business you must stay vigilant. There are always people with wrong intentions looking for the next way to compromise your business's information
- Perform annual or, better, quarterly vulnerability assessments to identify new risks. The ever-changing security environment is always creating new risks.
- Identify the new risks that apply to your business and fix them before someone else finds them.
- Get a formal Information Security Risk Assessment done every three years, which is the life cycle of most products these days.

Information security vulnerabilities and Risk Management

Objectives: At the end of this lesson you shall be able to

- describe security vulnerability
- describe the types of vulnerabilities
- explain how threats affect a system
- describe the vulnerability assessment tools and techniques
- describe the best practices to mitigate security vulnerabilities
- describe the relation between threat, vulnerability and risk
- describe the various threat agents
- describe the purpose of risk management
- describe the types of risk management
- explain the risks to ICT supply chain management.

Introduction

In computer security, a vulnerability is a software, hardware, procedural, or human weakness which allows an attacker to reduce a system's information assurance. A vulnerability is a weakness that may provide an attacker the open door he is looking for to enter a computer or network and have unauthorized access to resources within the environment. It represents the absence or weakness of a safety measure that could be exploited.

Threat, Threat Agent, Exploit, Risk and Vulnerability.

- A **threat** is any potential danger to information or systems. The threat is that someone, or something, will identify a specific vulnerability and use it against the company or individual.
- The entity that takes advantage of a vulnerability is referred to as a **threat agent**.
- An **exploit** is a means of taking advantage of the vulnerability and using it to take advantage of a system or network.
- A **risk** is the likelihood of a threat agent taking advantage of a vulnerability and the corresponding business impact.

In other words, a threat is what we're trying to protect against, a vulnerability is a weakness or gap in our protection efforts.

Why do Information Security Vulnerabilities exist?

Vulnerabilities exist because of exploits in code or networking protocols. Millions of lines of code are required to make an operating system, and sometimes vulnerabilities can be found within. If a firewall has several ports open, there is a higher likelihood that an intruder will use one to access the network in an unauthorized method. Vulnerability could be the result of "a flaw or weakness in hardware, software or process that exposes a system to compromise". It is the existence of a weakness, design, or implementation error that can lead to an unexpected, undesirable event compromising the security of the computer system, network, application, or protocol involved.

A Types of Technical Vulnerabilities

Most software security vulnerabilities fall into one of a small set of categories:

- buffer overflows
- unvalidated input
- race conditions
- access-control problems
- weaknesses in authentication, authorization, or cryptographic practices

Buffer Overflows

A buffer overflow, considered to be a major source of vulnerabilities, occurs when an application attempts to write data beyond the end (or, occasionally, before the beginning) of a buffer. Buffer overflows can cause applications to crash, can compromise data, and can provide an attack vector for further privilege escalation to compromise the system on which the application is running. Buffer overflow attacks generally occur by compromising either the stack, the heap, or both.

Unvalidated Input

As a general rule, you should check all input received by your program to make sure that the data is reasonable or valid. In cases where invalid data is allowed or accepted, a normal program attempting to read such a file would attempt to allocate a buffer of an incorrect size, leading to the potential for a heap overflow attack or other problem. For this reason, you must check your input data carefully. This process is commonly known as input validation or sanity checking.

Any input received by your program from an untrusted source is a potential target for attack. Hackers look at every source of input to the program and attempt to pass in malformed data of every type they can imagine. If the program crashes or otherwise misbehaves, the hacker then tries to find a way to exploit the problem.

Race Conditions

A race condition occurs when a pair of routine programming calls in an application do not perform in the sequential manner that was intended to as per rules. It is a timing event within software that can become a security vulnerability if the calls are not performed in the correct order. If the correct order of execution is required for the proper functioning of the program, this is a bug. If an attacker can take advantage of the situation to insert malicious code, change a filename, or otherwise interfere

with the normal operation of the program, the race condition is a security vulnerability. Attackers can sometimes take advantage of small time gaps in the processing of code to interfere with the sequence of operations, which they then exploit.

In software development, time of check to time of use (TOCTTOU or TOCTOU, pronounced "TOCK too") is a class of software bug caused by changes in a system between the checking of a condition (such as a security credential) and the use of the results. This is one example of a race condition.

Interprocess Communication(IPC)

Interprocess communication (IPC) is a set of programming interfaces that allow a programmer to coordinate activities among different program processes that can run concurrently in an operating system. This allows a program to handle many user requests at the same time. These messaging protocols used for interprocess communication are often vulnerable to attack.

Remote Procedure Call (RPC) is an interprocess communication mechanism that allows a program running on one host to run code on a remote host.

Insecure File Operations

In addition to time-of-check-time-of-use problems, many other file operations are insecure. Programmers often make assumptions about the ownership, location, or attributes of a file that might not be true. For example, you might assume that you can always write to a file created by your program. However, if an attacker can change the permissions or flags on that file after you create it, and if you fail to check the result code after a write operation, you will not detect the fact that the file has been tampered with.

B Types of Native Vulnerabilities

Examples of Native Vulnerabilities are:

- Vulnerabilities in the sandboxing mechanism which allow untrusted bytecode to circumvent the restrictions imposed by the security manager
- Vulnerabilities in the Java class library on which an application depends for its security

Understanding Security Vulnerabilities

Flaws in Software or Protocol Designs

Fundamental mistakes and oversight in Software design are the causes of design vulnerabilities. Design flaws result in software not being secure thus making it a high level vulnerability case.

Computer networks depend on protocols that specify the messages that are exchanged at runtime, their format and structure. Protocols are linked with different protocol stacks, e.g., TCP/IP, or different models, e.g., OSI, and many protocols with underspecified security are still present in practice. Some of the vulnerabilities arising due to flawed protocols are described below:

A. TCP/IP. The TCP/IP protocol stack has some weak points that allow:

- **Spoofing :** A spoofing attack is when a malicious party impersonates another device or user on a network in order to launch attacks against network hosts, steal data, spread malware, or bypass access controls. There are several different types of spoofing attacks that malicious parties can use to accomplish this. They are IP Address Spoofing Attacks, ARP (Address Resolution Protocol) Spoofing Attacks, DNS Server Spoofing Attacks, etc.
- **Telnet protocol :** Telnet can be used to administer systems running Microsoft Windows 2000 and Unix. When using the telnet client to connect from a Microsoft system to UNIX system and vice versa, user names and passwords are transmitted in clear text thus creating security vulnerability.
- **File Transfer Protocol (FTP) :** File Transfer Protocol allows users to connect to remote systems and transfer files back and forth. As part of establishing a connection to a remote computer, FTP relies on a user name and password combination for authentication. Use of FTP poses a security problem similar to use of the Telnet protocol because passwords typed to FTP are transmitted over the network in plain text, one character per packet. These packets can be intercepted.
- **Weaknesses in how protocols and software are implemented**

Even when a protocol is well designed, it can be vulnerable because of the way it is implemented. For example, a protocol for electronic mail may be implemented in a way that permits intruders to connect to the mail port of the victim's machine and fool the machine into performing a task not intended by the service. This type of vulnerability enables intruders to attack the victim's machine from remote sites without access to an account on the victim's system.

- Software may be vulnerable because of flaws that were not identified before the software was released. This type of vulnerability has a wide range of subclasses, which intruders often exploit using their own attack tools like race conditions in file access, non-existent checking of data content and size etc.

Weaknesses in System and Network Configurations

- System administrators and users may neglect to change the default settings in network configurations, or they may simply set up their system to operate in a way that leaves the network vulnerable.
- Asynchronous transfer mode (ATM). Security can be compromised by what is referred to as "manhole manipulation"-direct access to network cables and connections in underground parking garages and elevator shafts.
- Frame relay. Similar to the ATM problem.

- Device administration. Switches and routers are easily managed by an HTTP interface or through a command line interface. Coupled to the use of weak passwords (for example, public passwords), it allows anybody with some technical knowledge to take control of the device.
- Modems. A modem bypasses the "firewall" that protects a network from outside intruders. A hacker using a "war dialer" tool to identify the modem telephone number and a "password cracker" tool to break a weak password can gain access to the system.
- Weaknesses in Web or Cloud applications

There are several significant vulnerabilities that should be considered when an organization is ready to move their critical applications and data to a cloud computing environment, these vulnerabilities are described below :

A Session Riding and Hijacking

Session riding refers to the hackers sending commands to a web application on behalf of the targeted user by just sending that user an email or tricking the user into visiting a specially crafted website. Session riding deletes user data, executes online transactions like bids or orders, sends spam to an intranet system via internet and changes system as well as network configurations or even opens the firewall.

B Virtual Machine Escape

VM escape is a vulnerability that enables a guest-level VM to attack its host. Under this vulnerability an attacker runs code on a VM that allows an OS running within it to break out and interact directly with the hypervisor.

C Reliability and Availability of Service

The cloud storage infrastructure may go down for a considerable time, causing data loss and access issues with web services.

D Insecure Cryptography

Attackers' can decode any cryptographic mechanism or algorithm as main methods to hack them are discovered.

E Data Protection and Portability

Although the cloud services are offered based on a contract among client and a provider but what will happen when the contract is terminated and client doesn't want to continue anymore.

F Vendor Lock-in

This vulnerability occurs due to immature providers and new business models which raise the risk of failure and going out of the business.

G Internet Dependency

Cloud computing is an internet dependent technology where users are accessing the services via web browser. What if the internet is not available or service is down, what will happen to users systems and operations that are very critical and need to run 24 hours such as Healthcare and Banking systems.

Weaknesses in Online e-transactions

The tremendous increase in online transactions has been accompanied by an equal rise in the number and type of attacks against the security of online payment systems. Some of these attacks have utilized vulnerabilities that have been published in reusable third-party components utilized by websites, such as shopping cart software. Other attacks have used vulnerabilities that are common in any web application, such as SQL injection or cross-site scripting.

The common types of vulnerabilities in Online e-transactions are SQL injection, cross-site scripting, information disclosure, path disclosure, price manipulation, and buffer overflows.

Successful exploitation of these vulnerabilities can lead to a wide range of results. Information and path disclosure vulnerabilities will typically act as initial stages leading to further exploitation. SQL injection or price manipulation attacks could cripple the website, compromise confidentiality, and in worst cases cause the e-commerce business to shut down completely.

One of the main reasons for such vulnerabilities is the fact that web application developers are often not very well versed with secure programming techniques.

Browser Security and Role of cookies and pop-ups

Security vulnerabilities may allow a cookie's data to be read by a hacker, used to gain access to user data, or used to gain access (with the user's credentials) to the website to which the cookie belongs.

Pop-up ads or pop-ups

The "security" risks from popup windows are phishing, trapping to unwanted web sites etc.

Security holes in Browser, Web Applications, OS, and Smartphones: In security terminology, a hole refers to a software or operating system vulnerability that could be exploited to compromise the overall security of the computer system or network on which the hole resides. The three different kinds of vulnerabilities are:

- Operating system vulnerabilities are those affecting the Linux kernel; or components that ship with an operating system produced by Microsoft, Apple, or a proprietary Unix vendor, and defined as part of the operating system by the vendor.
- Browser vulnerabilities are those affecting components defined as part of a web browser. This includes web browsers that ship with operating systems, such as Windows Internet Explorer and Apple's Safari, along with third-party browsers, such as Mozilla Firefox and Google Chrome.
- Application vulnerabilities are those affecting all other components, including components published by operating system vendors and other vendors. Vulnerabilities in open source components that may ship with Linux distributions (such as the X Window System, the GNOME desktop environment, GIMP, and others) are considered application vulnerabilities.

Security holes in Web Applications

The following is a list of top 10 threats in the OWASP(Open Web Applications Security project) category.

Injection (Sql -> SQL Injection), Broken Authentication & Session Management, XSS (Cross Site Scripting), Insecure Direct Object Reference, Security Misconfiguration, Sensitive Data Exposure, Missing Function Level Access Control, Cross Site Request Forgery (CSRF Or XSRF), Using Components With Known Vulnerabilities and Unvalidated Redirect & Forwards.

Security holes in OS

Some of the vulnerabilities in UNIX OS are Setuid problems, Trojan Horses and Terminal Troubles

Some of the vulnerabilities in Windows OS are Passwords, Peer to Peer File sharing, Vulnerabilities in embedded automation features in Microsoft Outlook and Outlook Express that can allow execution of rogue code.

Some of the vulnerabilities in LINUX OS are missing permission checks, Uninitialized data, and Memory mismanagement

Some of the vulnerabilities in Smartphones are Data leakage resulting from device loss or theft, Unintentional disclosure of data, Attacks on decommissioned smartphones, Phishing attacks, Spyware attacks, Network Spoofing Attacks, Surveillance(user under surveillance) attacks, Diallerware attacks(Stealing money), Financial malware attacks(Stealing credentials) and Network congestion.

Vulnerability Assessment Tools and Techniques

Vulnerability Assessment is a Security Exercise that identifies weaknesses, identifies and enumerates vulnerabilities and reports on the discoveries about security liabilities within networks, applications and systems.

The Vulnerability Assessment detects vulnerabilities via:

- Security technologies
 - VA Scanners, Appliances and Software
- Remediation technologies
 - Patch Management Systems(WSUS, SCCM, LanDesk, VMWare Update manager)

Vulnerability Assessment involves mainly the following three steps:

- Information Gathering and Discovery which includes Network Scanning, Ports Scanning, Directory Services and DNS Zones and Registers.
- Enumeration which includes Hosts and OS, Ports, Services and their versions, information and SNMP communities
- Detection involving Identification of Weaknesses, Identification of Vulnerabilities, Report Generation and Use of remediation tools.

Vulnerability Assessment tools

Vulnerability Assessment tools detect, identify, measure the effect of the vulnerabilities found at various levels. Most Vulnerability Assessment tools are capable of scanning a number of network nodes, including networking and networked devices (switches, firewalls, printers, etc.) as well as server, desktop and portable computers.

Common Vulnerability Assessment Tools are Network Scanners, Host Scanners, Database Scanners, Web Application Scanners, Multilevel Scanners, Automated Penetration test tools and Vulnerability Scan Consolidators.

Techniques to Exploit Vulnerabilities

Vulnerabilities can be exploited for ex. by the use of packet sniffers. Other tools are used to construct packets with forged addresses; one use of these tools is to mount a denial-of-service attack in a way that hides the source of the attack. Intruders also "spoof" computer addresses, masking their real identity and successfully making connections that would not otherwise be permitted. In this way, they exploit trust relationships between computers.

The most common exploits occur by the use of Trojans, Viruses, Worms, Logic Bombs, Phishing, Forwarding and sharing Urban legends, Responding to Nigerian Scams etc.

Techniques to Fix the Vulnerabilities

Effective remediation demands continuous processes that together are called Vulnerability Management. The processes and related technology defined by vulnerability management help organizations efficiently find and fix network security vulnerabilities. Systematic use of these processes protects business systems from ever more frequent viruses, worms and other network-borne attacks.

Continuous Processes of Vulnerability Management involves Creating security policies & controls, Tracking inventory / categorizing assets, Scanning systems for vulnerabilities, Comparing vulnerabilities against inventory, Classifying risks, Pre-testing of patches, Applying patches and Re-scanning and confirming fixes. You can automate most of them now with security applications and Web-based services.

Best Practices and Guidelines to mitigate security Vulnerabilities

- 1 Initialize all variables before use
- 2 Validate all user input before use
- 3 Restrict administrative permissions on servers and databases
- 4 Handle errors and don't display system error messages to end users
- 5 Provide accounts with the least amount of permissions and privileges required
- 6 Don't store secrets (e.g. passwords, keys) in your code

- 7 Use tested, reliable libraries or modules for common functions (e.g. authentication, encryption, session tracking)
- 8 Secure login pages and pages protected by authentication with HTTPS
- 9 Ensure that server components (OS, software/apps) are up-to-date
- 10 Avoid installing unnecessary applications on production servers
- 11 Remove unused and backup pages from the web server
- 12 If possible, make code libraries and configuration files inaccessible from the web
- 13 Disable directory browsing
- 14 Avoid making operating system calls based on user input
- 15 Make use of the session tracking mechanism built into your development framework.

Risk is the combination of the probability of an event and its consequences. It refers to the likelihood of being targeted by a given attack.

Relationship between Threat, Vulnerability, and Risk

Before defining the relationship between threat, vulnerability and risk let us review the following terms:

Asset: In information Security, an asset is what we are trying to protect. It may be people, property or information.

Threat: Anything that can exploit a vulnerability, intentionally or accidentally and obtain, damage or destroy an asset.

Vulnerability: It refers to the weakness or gaps in our protection efforts.

Risk: When a threat exploits a vulnerability, it may cause loss, damage or destruction of an asset. This is called a Risk.

Risk is therefore the intersection of assets, threats and vulnerabilities.

ie. assets x threats x vulnerabilities = risk

Value of an asset

The value placed on information is relative to the parties involved, what work was required to develop it, how much it costs to maintain, what damage would result if it were lost or destroyed, what enemies would pay for it, and what liability penalties could be endured. If a company does not know the value of the information and the other assets it is trying to protect, it does not know how much money and time it should spend on protecting them. While assigning values to assets, one needs to consider certain issues which are stated as below.

- Cost to acquire or develop the asset
- Cost to maintain and protect the asset
- Value of the asset to owners and users

- Value of the asset to adversaries
- Value of intellectual property that went into developing the information
- Price that others are willing to pay for the asset
- Cost to replace the asset if lost or damaged
- Operational and production activities that are affected if the asset is unavailable
- Liability issues if the asset is compromised
- Usefulness and role of the asset in the organization

Understanding the value of an asset is the first step to understanding what security mechanisms should be utilized and what funds should go toward protecting it.

What Is a Threat Source/Agent?

A Threat Source or threat Agent is an entity with an intention and capability to cause impact.

Threat agents can take one or more of the following actions against an asset:

- Access - simple unauthorized access
- Misuse - unauthorized use of assets (e.g., identity theft, setting up a porn distribution service on a compromised server, etc.)
- Disclose - the threat agent illicitly discloses sensitive information
- Modify - unauthorized changes to an asset
- Deny access - includes destruction, theft of a non-data asset, etc.

The threat agents can be any of the following:

These individuals and groups can be classified as follows:

- Non-Target Specific: Non-Target Specific Threat Agents are computer viruses, worms, trojans and logic bombs.
- Employees: Staff, contractors, operational/maintenance personnel, or security guards who are annoyed with the company.
- Organized Crime and Criminals: Criminals target information that is of value to them, such as bank accounts, credit cards or intellectual property that can be converted into money. Criminals will often make use of insiders to help them.
- Corporations: Corporations are engaged in offensive information warfare or competitive intelligence. Partners and competitors come under this category.
- Human, Unintentional: Accidents, carelessness.
- Human, Intentional: Insider, outsider.
- Natural: Flood, fire, lightning, meteor, earthquakes.

Risk Controls

If the mitigation of risk is the central focus of Information Security, Controls are the primary tools to achieve this goal. A control is any device or process that is used to reduce risk.

Basically the three types of controls are:

- 1 **Administrative** : Administrative controls are the actions that people take. Administrative controls are the process of developing and ensuring compliance with policy and procedures
- 2 **Technical or Logical** : These are the virtual, application and technical controls (systems and software), such as firewalls, anti virus software, encryption and maker/checker application routines. Technical controls are carried out or managed by computer systems.
- 3 **Activity phase controls** can be either technical or administrative and are classified as follows based on the level of risk mitigation:
 - Preventative controls exist to prevent the threat from coming in contact with the weakness. These are controls that prevent the loss or harm from occurring. For example, a control that enforces segregation of responsibilities (one person can submit a payment request, but a second person must authorize it), minimizes the chance an employee can issue fraudulent payments.
 - Detective controls exist to identify that the threat has landed in our systems. These controls monitor activity to identify instances where practices or procedures were not followed. For example, a business might reconcile the general ledger or review payment request audit logs to identify fraudulent payments.
 - Corrective controls exist to mitigate or lessen the effects of the threat being manifested. Corrective controls restore the system or process back to the state prior to a harmful event. For example, a business may implement a full restoration of a system from backup tapes after evidence is found that someone has improperly altered the payment data.
 - Compensating controls are alternate controls designed to accomplish the intent of the original controls as closely as possible, when the originally designed controls cannot be used due to limitations of the environment.

Risk likelihood

Risk likelihood is a rough measure of how likely this particular vulnerability is to be uncovered and exploited by an attacker. It is not necessary to be over-precise in this estimate. Generally, identifying whether the likelihood is low, medium, or high is sufficient.

There are a number of factors that can help determine the likelihood. The first set of factors are related to the threat agent involved. The goal is to estimate the likelihood of a successful attack from a group of possible attackers. Note that there may be multiple threat agents that can exploit a particular vulnerability, so it's usually best to use the worst-case scenario. For example, an insider may be a much more likely attacker than an anonymous outsider, but it depends on a number of factors.

The first set of factors are related to the threat agent involved. The goal here is to estimate the likelihood of a successful attack by this group of threat agents. The second factor to be taken into account is the Motive behind the attacks. The Access or resources required and the size of the group of threat agents are the other factors to be considered.

The next set of factors are related to the vulnerability involved. The goal here is to estimate the likelihood of the particular vulnerability involved being discovered and exploited. This takes into account the ease of discovery, the ease of exploit, the awareness to this group of threat agents, the likelihood of detection of this exploit.

Factors for Estimating Impact

When considering the impact of a successful attack, it's important to realize that there are two kinds of impacts. The first is the "technical impact" on the application, the data it uses, and the functions it provides. The other is the "business impact" on the business and company operating the application.

Ultimately, the business impact is more important. However, you may not have access to all the information required to figure out the business consequences of a successful exploit. In this case, providing as much detail about the technical risk will enable the appropriate business representative to make a decision about the business risk.

Technical impact can be broken down into factors aligned with the traditional security areas of concern: confidentiality, integrity, availability, and accountability. The goal is to estimate the magnitude of the impact on the system if the vulnerability were to be exploited. The issues to be considered are Loss of confidentiality, Loss of integrity, Loss of availability and Loss of accountability.

The factors to be considered for assessing the business impact are financial damage, Reputation damage, Non-compliance to policies and Privacy violation.

Risk Control Effectiveness: The risk control effectiveness depends on the Number of systemic risks identified, Percentage of process areas involved in risk assessments, Percentage of key risks mitigated and Percentage of key risks monitored among many factors involved.

Risk Management

Risk Management and Risk Assessment are major components of Information Security Management (ISM).

Risk management is the process of identifying vulnerabilities and threats to the information resources used by an organization and deciding what countermeasures to take in reducing the value of the asset.

The process of risk management is an ongoing, iterative process. It must be repeated indefinitely. The business environment is constantly changing and new threats and vulnerabilities emerge every day. Second, the choice of countermeasures (controls) used to manage risks must strike a balance between productivity, cost, effectiveness of the countermeasure, and the value of the informational asset being protected.

On the contrary, Risk Assessment is executed at discrete time points (ex. once a year, on demand, etc.) and until the performance of the next assessment - provides a temporary view of assessed risks.

Purpose of Risk Management

The principle reason for managing risk in an organization is to protect the mission and assets of the organization. Therefore, risk management must be a management function rather than a technical function. Understanding risk, and in particular, understanding the specific risks to a system allow the system owner to protect the information system commensurate with its value to the organization. The fact is that all organizations have limited resources and risk can never be reduced to zero. So, understanding risk, especially the magnitude of the risk, allows organizations to prioritize scarce resources.

Risk Assessment (Phases)

The purpose of assessing risk is to assist management in determining where to direct resources. There are four basic strategies for managing risk: mitigation, transference, acceptance and avoidance.

Mitigation

Mitigation is the most commonly considered risk management strategy. Mitigation involves fixing the flaw or providing some type of compensatory control to reduce the likelihood or impact associated with the flaw. A common mitigation for a technical security flaw is to install a patch provided by the vendor. Sometimes the process of determining mitigation strategies is called control analysis.

Transference

Transference is the process of allowing another party to accept the risk on your behalf. This is not widely done for IT systems, but everyone does it all the time in their personal lives. Car, health and life insurance are all ways to transfer risk. In these cases, risk is transferred from the individual to a pool of insurance holders, including the insurance company. Note that this does not decrease the likelihood or fix any flaws, but it does reduce the overall impact (primarily financial) on the organization.

Acceptance

Acceptance is the practice of simply allowing the system to operate with a known risk. Many low risks are simply accepted. Risks that have an extremely high cost to mitigate are also often accepted. Beware of high risks being accepted by management. Ensure that this strategy is in writing and accepted by the manager(s) making the decision. Often risks are accepted that should not have been accepted, and then when the penetration occurs, the IT security personnel are held responsible. Typically, business managers, not IT security personnel, are the ones authorized to accept risk on behalf of an organization.

Avoidance

Avoidance is the practice of removing the vulnerable aspect of the system or even the system itself. For instance, during a risk assessment, a website was uncovered that

let vendors view their invoices, using a vendor ID embedded in the HTML file name as the identification and no authentication or authorization per vendor. When notified about the web pages and the risk to the organization, management decided to remove the web pages and provide vendor invoices via another mechanism. In this case, the risk was avoided by removing the vulnerable web pages.

Types of Risk Assessment

Quantitative Risk Assessment

Quantitative risk assessment draws upon methodologies used by financial institutions and insurance companies. By assigning values to information, systems, business processes, recovery costs, etc., impact, and therefore risk, can be measured in terms of direct and indirect costs.

But it is not commonly used to measure risk in information systems because

- 1 The difficulties in identifying and assigning a value to assets, and
- 2 The lack of statistical information that would make it possible to determine frequency.

Thus, most of the risk assessment tools that are used today for information systems are measurements of qualitative risk.

Qualitative Risk Assessment

Qualitative risk assessments assume that there is already a great degree of uncertainty in the likelihood and impact values and defines them, and thus risk, in somewhat subjective or qualitative terms. Similar to the issues in quantitative risk assessment, the great difficulty in qualitative risk assessment is defining the likelihood and impact values.

Moreover, these values need to be defined in a manner that allows the same scales to be consistently used across multiple risk assessments. Qualitative risk assessments typically give risk results of "High", "Moderate" and "Low". However, by providing the impact and likelihood definition tables and the description of the impact, it is possible to adequately communicate the assessment to the organization's management.

The risk assessment includes the following actions and activities:

- Identification of assets;
- Identification of legal and business requirements that are relevant for the Identified assets
- Valuation of the identified assets, taking account of the identified legal and
- Business requirements and the impacts of a loss of confidentiality, integrity and availability
- Identification of significant threats to, and vulnerabilities of, the identified assets;
- Assessment of the likelihood of the threats and vulnerabilities to occur;

- Calculation of risk;
- Evaluation of the risks against a predefined risk scale.

Risks to the ICT supply chain management

Risks to the ICT supply chain arise from the loss of confidentiality, integrity, or availability of information or information systems and reflect the potential adverse impacts to organizational operations (including mission, functions, image, or reputation), organizational assets, individuals, and other organizations. Such risks can take the form of sloppy software development, inadequate testing of products before they are shipped, and using the least expensive parts even if that means they may not be authentic.

Supply chain risk management (SCRM) is the process of understanding these risks, their business impacts, and how to manage them by mitigating supply chain weaknesses and exploits throughout the system lifecycle.

Directory services

Objectives: At the end of this lesson you shall be able to

- describe directory and directory service
- describe the benefits of directory services
- mention the various implementations of directory services
- describe the logical and physical structure of active directory
- describe global catalog and group policy.

Introduction

Directories and directory services

A **directory** is a collection or list of data. Real world examples are telephone books, land registers and listings of works. All these examples have the purpose of preserving information and making it available on demand to the concerned persons.

Within information technology the term directory is used for a special kind of data storage. It allows the structured storage and efficient retrieval of objects which are often derived from the real world (e.g. persons, IT equipment). The main characteristic of this storage is that all data is stored in so called entries. The set of entries within a directory forms a tree (hierarchical database).

Directory Services

A **directory service** is a shared information infrastructure for locating, managing, administering, and organizing common items and network resources. These can include volumes, folders, files, printers, users, groups, devices, telephone numbers and other objects. A directory service is a solution which offers users access to the information stored in the directory through a well-defined interface. If a network is used, an appropriate protocol has to be defined for this purpose. The Light weight Directory Access Protocol(LDAP) is such a protocol.

Benefits of using directory services:

- 1 **Resource management:** The Network Administrator may use a single tool to manage network resources-- user accounts, servers, drives, files, printers, etc. These

Effective ICT SCRM requires processes, procedures, and tools that allow organizations to apply SCRM principles consistently across all ICT systems. One such principle is to minimize the risk of counterfeit parts since they may lead to unpredictable behavior, early failures, or worse. It therefore becomes necessary to distinguish counterfeit parts from authentic parts.

A structured language to express these characteristics is needed, such that all members of a supply chain can communicate about them, and which can be used to alert others about counterfeits or express the criteria for legitimate items. A structured language to describe these observable attributes of both legitimate and illegitimate components is one tool for reducing supply chain risk.

are displayed in a simple tree structure, so they are easy to locate and manage. Within this structure, each resource is displayed by an icon called an object. By selecting an object, its settings are available, and the Network Administrator may modify them as he sees fit.

2 Users: You may use a single log on to access resources to which you've been granted rights. Rather than having to log in to every server or authenticate to every printer or other device, the directory contains this information. You log on once and you can do whatever the Network Administrator has allowed via rights granted to you.

3 Security : Having only one domain means better security through a single security policy and a single set of administrators. If you have multiple domains and forests, each has its own administrator. One weak but trusted domain exposes all the other forests and domains. With only a single domain, it's also far easier to enforce an organization-wide security policy

4 Single platform: A single directory service or Global Catalog (GC) means a single platform for all other directory-ware services, including monitoring and messaging.

5 Faster deployment: starts in an organization with just a single domain and shared account database solutions need only be deployed once, which means company-wide deployments are much faster than if the organization has multiple and separate domains.

- 6 Single management:** infrastructure-Having a single management infrastructure means there is just one infrastructure for all other directory services tasks, such as software deployment, inventory, and object management sharing and delegation (such as for user accounts).
- 7 Single Group Policy container (GPC):** With a single GPC, management policies need to be defined only once, and can be used throughout the entire enterprise without the need to manually export and import Group Policy Objects (GPOs).
- 8 Backup and recovery:** Having only a single domain means better resiliency because every location has a full domain backup.
- 9 Less hardware:** In an organization with multiple domains, every location needs two domain controllers (DCs). With a single domain, each location needs only a single DC because if the local DC fails, the locations can use hub DCs. Reduced hardware also means fewer licenses, less management software, and less overhead for server management. There's also no need to back up remote DCs because the remote DCs just hold the same information as the central DCs-assuming the DCs only perform directory services.

Implementations of directory services

Directory services were part of an Open Systems Interconnection (OSI) initiative to get everyone in the industry to agree to common network standards to provide multi-vendor interoperability. In the 1980s, the International Telecommunication Union (ITU) and the International Organization for Standardization (ISO) came up with a set of standards - X.500, for directory services. The protocol decided upon is the Light weight Directory Access Protocol, LDAP, which is based on the directory information services of X.500, but uses the TCP/IP stack and a string encoding scheme of the X.500 protocol DAP.

Among the LDAP/X.500 based implementations are:

- Active Directory:** Microsoft's modern directory service for Windows, originating from the X.500 directory, created for use in Exchange Server, first shipped with Windows 2000 Server and is supported by successive versions of Windows.
- Apache Directory Server:** Directory service written in Java, supporting LDAP, Kerberos 5 and the Change Password Protocol. LDAPv3 certified. The Apache Directory Server is also a top level project of the Apache Software Foundation.
- eDirectory:** This is NetIQ's implementation of directory services. It supports multiple architectures including Windows, NetWare, Linux and several flavours of Unix and has long been used for user administration, configuration management, and software management. eDirectory has evolved into a central component in a broader range of Identity management products. It was previously known as Novell Directory Services.

- Red Hat Directory Server:** Red Hat released a directory service, that it acquired from AOL's Netscape Security Solutions unit, as a commercial product running on top of Red Hat Enterprise Linux called Red Hat Directory Server and as the community supported 389 Directory Server project.
- Oracle Internet Directory:** (OID) is Oracle Corporation's directory service, which is compatible with LDAP version 3.
- Sun Java System Directory Server:** Sun Microsystems' current directory service offering.
- OpenDS:** An open source directory service implementation from scratch in Java, backed by Sun Microsystems.
- IBM Tivoli Directory Server:** It is a customized build of an old release of OpenLDAP.
- Windows NT Directory Services (NTDS),** later renamed Active Directory, replaces the former NT Domain system.
- OpenLDAP :** It supports all current computer architectures, including Unix and Unix derivatives, Linux, Windows, z/OS, and a variety of embedded/realtime systems.

There are also plenty of open-source tools to create directory services, including OpenLDAP and the Kerberos protocol, and Samba software.

Active Directory

- Active Directory (AD)** is a directory service that Microsoft developed for Windows domain networks and is included in most Windows Server operating systems as a set of processes and services.
- An AD domain controller authenticates and authorizes all users and computers in a Windows domain type network-assigning and enforcing security policies for all computers and installing or updating software. For example, when a user logs into a computer that is part of a Windows domain, Active Directory checks the submitted password and determines whether the user is a system administrator or normal user.[3]
- Active Directory makes use of Lightweight Directory Access Protocol (LDAP) versions 2 and 3, Microsoft's version of Kerberos, and DNS.
- Active Directory also makes user management easier as it acts as a single repository for all of this user and computer related information.
- AD uses LDAP as its access protocol.
- AD relies on DNS as its locator service, enabling clients to locate domain controllers through DNS queries.

Logical Structure of Active Directory

Active Directory is a distributed database that stores and manages information about network resources, as well as application-specific data from directory enabled applications.

Active Directory allows administrators to organize elements of a network (such as users, computers, devices, and so on) into a hierarchical containment structure.

In Active Directory, resources are organized in a logical structure, and this grouping of resources logically enables a resource to be found by its name rather than by its physical location.

Benefits of AD Logical Structure

- Logical Structure provides more network security by means of providing access to resources to only specified groups (OU).
- Logical structure simplified the network management by administration, configuration and control of the network.
- The relationship between the logical structure of domains and forests simplifies resource sharing across an organization.
- As logical structure provides simplified network management, it reduces the load on network resources and lower the total cost of ownership.

Components of AD Logical Structure

The logical structure components have relationship with each other so it manage to control access to stored data and finds how the data will be managed between different domains in a forest.

- Objects: like a user, computer, group, printer etc...
- Organizational Units - like any folder but in control of Active Directory
- Domains - Logical boundaries for objects
- Trees - Logical boundary for multiple domains
- Forests - Logical boundary for multiple trees

Overall, one physical machine running as a Microsoft Domain controller can control all these logical divisions with the help of 'A Operation Master' dedicated to perform specific tasks.

The top-level container is the forest. A forest is a collection of trees that share a common global catalog, directory schema, logical structure, and directory configuration. The forest represents the security boundary within which users, computers, groups, and other objects are accessible.

Within forests are domains. A domain is defined as a logical group of network objects (computers, users, devices) that share the same active directory database.

Within domains are organizational units. OUs can provide hierarchy to a domain, ease its administration, and can resemble the organization's structure in managerial or geographical terms. OUs can contain other OUs-domains are containers in this sense. The OU is the level at which administrative powers are commonly delegated, but delegation can be performed on individual objects or attributes as well.

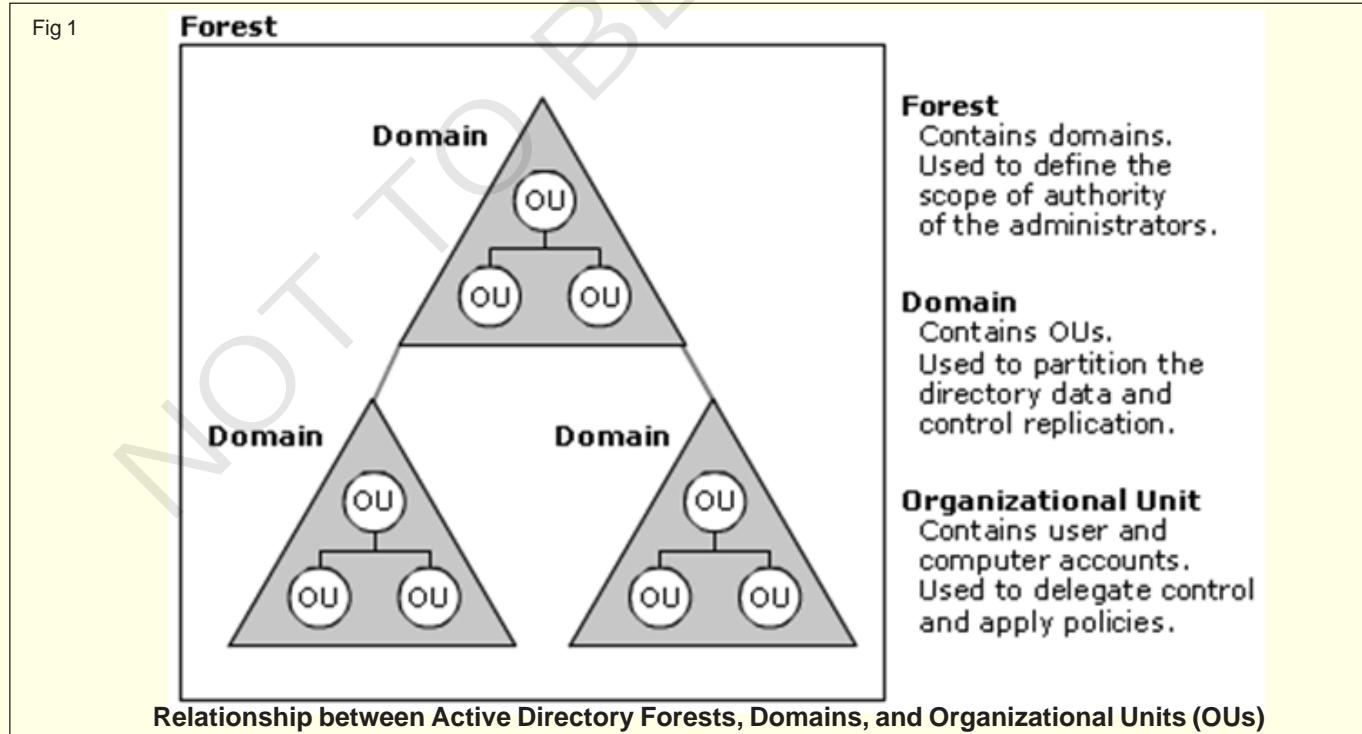
This is called the logical model because it is independent of the physical aspects of the deployment, such as the number of domain controllers required within each domain and network topology.

Figure 1 shows the relationship between forests, domains, and organizational units.

Figure 1 Relationship between Active Directory Forests, Domains, and Organizational Units(OUs)

The Physical Structure of an Active Directory

The Active Directory physical structure checks when and where logon and replication traffic occurs. The physical structure of Active Directory contains all the physical subnets present in network like domain controllers and replication between domain controllers.



The physical structure of Active Directory:

- **Domain Controllers:** These computers run Microsoft Windows Server 2003/2000, and Active Directory. Every Domain Controller performs specific functions like replication, storage and authentication. It can support maximum one domain. It is always advised to have more than one domain controller in each domain.
- **Active Directory Sites:** These sites are collection of well-connected computers. The reason why we create site is domain controllers can communicate frequently within the site. This way it minimizes the latency within site say changes made on one domain controller to be replicated to other domain controllers. The other reason behind creating a site is to optimize bandwidth between domain controllers which are located in different locations.

All IP subnets who share the common Local Area Network (LAN) connectivity without knowing the actual physical location of computers is called site.

A global catalog is a data storage source containing partial representations of objects found in a multidomain **Active Directory Domain Services** (ADDS) forest. The global catalog is stored on domain controllers specifically assigned as global catalog servers. It can locate objects in any domain without knowing the actual domain name.

Searches that are directed to the global catalog are faster because they do not involve referrals to different domain controllers.

Organizing resources in OU

OUs are the primary method for organizing user, computer, and other object information into a more easily understandable layout. The organization has a root organizational unit where three nested organizational units are placed. This nesting enables the organization to distribute users across multiple containers for easier viewing and administration of network resources.

OUs can be further subdivided into resource OUs for easy organization and delegation of administration. Far-flung offices could have their own OUs for local administration as well. It is important to understand, however, that an OU should be created only if the organization has a specific need to delegate administration to another set of administrators. If the same person or group of people administer the entire domain, there is no need to increase the complexity of the environment by adding OUs. In fact, too many OUs can impact group policies, logons, and other factors.

OUs can be structured to allow for separate departments to have various levels of administrative control over their own users. For example, a secretary in the Engineering department can be delegated control of resetting passwords for users within his own OU. Another advantage of OU use in these situations is that users can be easily dragged and dropped from one OU to another. For example, if users are moved from one department to another, moving them into their new department's OU is extremely simple.

It is important to keep in mind that OU structure can be modified on the fly any time an administrator feels fit to make structural changes. This gives Active Directory the added advantage of making changes any time.

Group Policy is a feature of the Microsoft Windows NT family of operating systems that control the working environment of user accounts and computer accounts. Group Policy provides the centralized management and configuration of operating systems, applications, and users' settings in an Active Directory environment. A version of Group Policy called Local Group Policy ("LGPO" or "LocalGPO") also allows Group Policy Object management on standalone and non-domain computers. Group Policy is one of the top reasons to deploy Active Directory because it allows you to manage user and computer objects.

Group Policy, in part, controls what users can and cannot do on a computer system, for example: to enforce a password complexity policy that prevents users from choosing an overly simple password, to allow or prevent unidentified users from remote computers to connect to a network share, to block access to the Windows Task Manager or to restrict access to certain folders. A set of such configurations is called a Group Policy Object (GPO).

Active Directory Backup and Restore

Active Directory is one of the most critical components of your infrastructure. If it goes down, your network is rendered useless. Therefore, to ensure business continuity and compliance, you need to have a solid backup and recovery plan in place for Active Directory.

Where a Group Policy Preference Settings is configured and there is also an equivalent Group Policy Setting configured, then the value of the Group Policy Setting will take precedence. Group policy is security of domain. Backup utility will automatically locate and include them when you back up system state.

Access Control, Audit and testing

Objectives: At the end of this lesson you shall be able to

- **describe the purpose and steps in access control**
- **describe the various layers in access control**
- **describe the access control mechanisms**
- **classify information for security**
- **list the access control protocols**
- **describe security audit**
- **describe the important of security audits**
- **describe the process of performing audit security**
- **describe vulnerability assessment and penetration testing**
- **mention the various types of security audit tools.**

Introduction: Access to protected information must be restricted to people who are authorized to access the information. The purpose of Access Control is to limit the actions or operations that a legitimate user of a computer system can perform, as well as what programs executing on behalf of the users are allowed to do. In this way access control seeks to prevent activity that could lead to a breach of security.

The sophistication of the access control mechanisms should be proportional to the value of the information being protected - the more sensitive or valuable the information the stronger the control mechanisms need to be. The foundation on which access control mechanisms are built start with identification and authentication.

Access control is generally achieved in three steps: Identification, Authentication, and Authorization.

- Identification is an assertion of who someone is or what something is. A user identifies himself by entering by entering that username.
- Authentication is the process of verifying the user's identity. Authentication is one of the five pillars of information assurance (IA). The other four are integrity, availability, confidentiality and nonrepudiation.

Entering the password is a common method for authentication. But password-based authentication is not suitable for use on computer networks. Passwords sent across the networks can be intercepted and subsequently misused by hackers. In addition to the security concern, password based authentication is inconvenient as the user has to enter the password each time the network service is to be accessed. The weakness in this system for transactions that are significant (such as the exchange of money) is that passwords can often be stolen, accidentally revealed, or forgotten.

Authorization : After a person, program or computer has successfully been identified and authenticated then it must be determined what informational resources they are permitted to access and what actions they will be allowed to perform (run, view, create, delete, or change). This is called Authorization. Authorization to access information and other computing services begins with administrative policies and procedures. The policies prescribe what information and computing services can be accessed, by whom, and under what conditions.

Authentication Methods

- The risk of eavesdropping can be managed by using digests for authentication. The connecting party sends a value, typically a hash of the client IP address, time stamp, and additional secret information. Because this hash is unique for each accessed URI, no other documents can be accessed nor can it not be used from other IP address without detection. The password is also not vulnerable to eavesdropping because of the hashing. The system is, however, vulnerable to active attacks such as the -man-in-the middle attack.
- One-time passwords: To avoid the problems associated with password reuse, one-time passwords were developed. There are two types of one-time passwords, a challenge-response password and a password list.
- Public -key cryptography: Public key cryptography is based on very complex mathematical problems that require very specialized knowledge. Public key cryptography makes use of two keys, one private and the other public. The two keys are linked together by way of an extremely complex mathematical equation. The private key is used to decrypt and also to encrypt messages between the communicating machines. Both encryption and verification of signature is accomplished with the public key.
- Zero -knowledge proofs: Zero-knowledge proofs make it possible for a Host to convince another Host to allow access without revealing any "secret information". The hosts involved in this form of authentication usually communicate several times to finalize authentication.
- Use of digital certificates issued and verified by a Certificate Authority (CA) as part of a public key infrastructure is considered another standard way to perform authentication on the Internet.
- Another method of authentication, biometrics, depends on the user's presence and biological makeup (i.e., retina or fingerprints). This technology makes it more difficult for hackers to break into computer systems. Using special protocols for authentication like Secure Sockets Layer (SSL), IP SEC, Secure Shell (SSH), Kerberos authentication and Extensible Authentication Protocol (EAP) etc.

A security administrator maintains a database of authorizations based on the security policy of the organization. The reference monitor consults an authorization database in order to determine if the user attempting to do an operation is actually authorized to perform that operation. Auditing monitors and keeps a record of relevant activity in the system.

It is important to make a clear distinction between authentication and access control. Correctly establishing the identity of the user is the responsibility of the authentication service. Access control assumes that the authentication of the user has been successfully verified prior to enforcement of access control via a reference monitor.

Successive Layers of Access Control

Access Control is implemented in successive layers and each layer builds upon the one that precedes it. Organizations can, optionally, uptake the various layers depending on the degree of automation and scalability they wish to build upon the existing Function and Data Security models. There can be various models to implement this. As an example, the Oracle User Management has six layers of access control. (Refer Fig 1) The Core Security layers include:



- Function Security
- Data Security

The next four layers are part of Oracle User Management:

- Role-Based Access Control
- Delegated Administration
- Registration Processes
- Self Service and Approvals

In general, Access Control begins with basic system administration tasks, progresses to more distributed, local modes of administration, and ultimately enables users to perform some basic, predefined registration tasks on their own. Table 1. illustrates how the layers build upon each other.

The Security and Data Security mechanisms constitute the base layers of the security system, and contain the traditional system administrative capabilities. They limit the scope of User Management to basic system administration by granting access to specific menus.

Table 1

Layer of Access	Level of Administration
Self Service and Approvals	End Users
Registration Processes	
Delegated Administration	Local Administration
Role Based Access Control	
Data Security	System Administrator
Function Security	

Local Administrators: When Role-Based Access Control and Delegated Administration are added to the Data Security and Function Security layers, system administration tasks can be distributed to local administrators who manage a subset of the organization's users.

End Users: Registration Processes and Self Service and Approvals distribute system administration further by automating some registration tasks so that end users can perform them. End users can perform the tasks of Obtaining new User Accounts, Request additional access to the system and reset passwords.

Self Service and Approvals: After the registration processes have been configured as per requirements, individuals can subsequently perform self-service registration tasks, such as obtaining new user accounts or requesting additional access to the system. In addition, organizations can use the Oracle Approvals Management engine to create customized approval routing for these requests.

There are three types of Preventive controls :

Administrative controls

- Policies/Procedures: to identify the ways in which processes must be performed. This must go hand in hand with training, detective controls and audits.

Physical Controls

- Using Biometric sensors, Smart cards etc.

Technical (Logical)Controls

- Encryption
- Passwords and Tokens
- Biometrics
- O.S. and Application Controls
- Identification and Authorization Technologies

Access Control Mechanisms

The following are the models/mechanisms for access control. Each of the above Access Models has its own advantages and disadvantages. The selection of the appropriate Access Model by an organization should be done by considering various factors such as type of business, no of users, organization's security policy etc.

Table 2

Control Service	Description
Preventive	Keep Undesirable Things from Happening
Detective	Identify Undesirable things that have taken place
Corrective	Correct Undesirable things that have taken place
Deterrent	Discourage Security Violations from taking place
Recovery	Restore Resources or Capabilities after a Violation or Accident
Compensation	Provide Alternatives to other Controls

- **Role Based Access Control (RBAC)**

Access decisions are based on an individual's roles and responsibilities within the organization or user base. RBAC is also known as non-discretionary Access Control because the user inherits privileges that are tied to his role. The user does not have a control over the role that he will be assigned.

- **Discretionary Access Control (DAC)**

As the name suggests, this access control model is based on a user's discretion. i.e, the owner of the resource can give access rights on that resource to other users based on his discretion. Access Control Lists (ACLs) are a typical example of DAC. Specifying the "rwx" permissions on a Unix file owned by you is another example of DAC. Most of the operating systems including windows, flavours of Unix are based on DAC Model.

- **Mandatory Access Control (MAC)**

In this Model, users/owners do not enjoy the privilege of deciding who can access their files. Here the operating system is the decision maker overriding the user's wishes. In this model every Subject (users) and Object (resources) are classified and assigned with a security label. The security labels of the subject and the object along with the security policy determine if the subject can access the object. The rules for how subjects access objects are made by the security officer, configured by the administrator, enforced by the operating system, and supported by security technologies.

This is a stricter and rather static Access Control model as compared to DAC and is mostly suited for military organizations where data classification and confidentiality is of prime importance. Special types of the Unix operating systems are based on MAC model.

- **Attribute Based Access Control (ABAC)**

This is to grant or deny user requests based on arbitrary attributes of the user, arbitrary attributes of the object, and environment conditions that may be more relevant to the policies at hand.

Password Cracking Methods And Their Counter measures

There are number of methods used by hackers to hack the accounts or steal personal information. Some of the most commonly used methods to crack passwords and their counter measures are as follows:

1 BruteForce Attack

Any password can be cracked using Brute-force attack. Brute-force attacks try every possible combinations of numbers, letters and special characters until the right password is match. Brute-force attacks can take very long time depending upon the complexity of the password. depending on the speed of computer and complexity of the password.

Countermeasure: Use long and complex passwords. Try to use combination of upper and lowercase letters along with numbers. Brute-force attack will take very long time to crack such complex and long passwords. You may even keep changing the passwords frequently.

2 Social Engineering

Social engineering is process of manipulating someone to trust you and get information from them. For example, if the hacker was trying to get the password of a co-workers or friends computer, he could call him pretending to be from the IT department or a bank and simply ask for his login or credit card details.

Countermeasure: Never ever give your sensitive information like credit card details on phone.

3 Rats And Keyloggers

In **keylogging or rating** the hacker sends keylogger or Rat to the victim. This allows hacker to monitor everything victim do on his computer. Every keystroke is logged including passwords. Moreover hacker can even control the victim's computer too.

Countermeasure: Never login to your bank account from the cyber cafe or someone else's computer. If it is very important, use on-screen or virtual keyboard while tying the login. Use latest anti-virus software and keep the definitions updated.

4 Phishing

Phishing is the most easiest and popular hacking method used by hackers to get someone account details. In Phishing attack hacker send fakepage of real website like facebook, gmail to victim. When someone logs in through that fake page his details is sent to the hacker. This fake pages can be easily created and hosted on free web-hosting sites.

Countermeasure: Phishing attacks are very easy to avoid. The url of this phishing pages are different from the real one. For example URL of phishing page of facebook might look like facbbook.com (As you can see There are two "b"). Always make sure that websites url is correct.

5 Rainbow Table

A Rainbow table is a huge pre-computed list of hashes for every possible combination of characters. A password hash is a password that has gone through a mathematical algorithm such as md5 and is transformed into something which is not recognizable. A hash is a one way encryption so once a password is hashed there is no way to get the original string from the hashed string.

Countermeasure: Make sure you choose password that is long and complex. Creating tables for long and complex password takes a very long time and a lot of resources.

6 Guessing

This is a simple method to help you get someone's password within seconds. If hacker knows you, he can use information he knows about you to guess your password. Hacker can also use combination of Social Engineering and Guessing to acquire your password.

Counter measure: Don't use your name, surname, phone number or birth date as your password. Try to avoid creating password that relates to you. Create complex and long password with combination of letters and numbers.

Security classification for information

Not all information is equal and so not all information requires the same degree of protection. This requires information to be assigned a security classification.

The first step in information classification is to identify a member of senior management as the owner of the particular information to be classified. Next, develop a classification policy. The policy should be able to:

- Storing information
- Transmitting information
- Describe different classification labels,
- Define the criteria for information to be assigned a particular label, and
- List the required security controls for each classification.
- Disposing of unneeded information
- Protecting the integrity of information
- Allowing appropriate access and disclosure
- Establishing accountability.

Some factors that influence which classification information should be assigned include:

- How much value that information has to the organization
- How old the information is and
- Whether or not the information has become obsolete.
- Laws and other regulatory requirements

The Business Model for Information Security enables security professionals to examine security from systems perspective, creating an environment where security can be managed holistically, allowing actual risks to be addressed.

The type of information security classification labels selected and used will depend on the nature of the organization, with examples being:

- In the business sector, labels such as: Public, Sensitive, Private and Confidential.
- In the government sector, labels such as: Unclassified, Sensitive But Unclassified, Restricted, Confidential, Secret, Top Secret and their non-English equivalents.
- In cross-sectoral formations, the Traffic Light Protocol, which consists of: White, Green, Amber, and Red.

All employees in the organization, as well as business partners, must be trained on the classification scheme and understand the required security controls and handling procedures for each classification. The classification of a particular information asset that has been assigned should be reviewed periodically to ensure the classification is still appropriate for the information and to ensure the security controls required by the classification are in place.

Declassifying and downgrading

Information must be classified or designated only for the time it requires protection, after which it is to be declassified or downgraded. This is because the classified or designated information will lose its sensitivity with the passage of time or the occurrence of specific events. This process contributes to the overall integrity of the security system, and ensures that information is made available quickly and informally to interested members of the public.

Access Control Administration

Access control administration can be done in two ways.

- **Centralized :** Here one entity (dept or an individual) is responsible for overseeing access to all corporate resources. This type of administration provides a consistent and uniform method of controlling users access rights. Example: RADIUS, TACACS and Diameter
- Decentralized

Access Control / Data Collection Protocols: AAA (RADIUS, Diameter, and TACACS+)

RADIUS, Diameter, and TACACS+ are three protocols for carrying Authentication, Authorization, and Accounting (AAA) information between a Network Access Server (NAS) that wants to authenticate its links or end users.

RADIUS

The Remote Authentication Dial-In User Service (RADIUS) is a client/server security protocol created by Lucent InterNetworking Systems. RADIUS is an Internet draft standard protocol. User profiles are stored in a central location, known as the RADIUS server. RADIUS clients (such as a PortMaster communications server) communicate with the RADIUS server to authenticate users. The server specifies back to the client what the authenticated user is authorized to do. Although the term RADIUS refers to the network protocol that the client and server use to communicate, it is often used to refer to the entire client/server system.

Diameter: Diameter is an authentication, authorization, and accounting protocol for computer networks. It evolved from and replaces the much less capable RADIUS protocol that preceded it. Diameter Applications extend the base protocol by adding new commands and/or attributes. It provides better, better transport, better security, better proxying, better session control and better interoperability when compared to RADIUS.

TACACS: Terminal Access Controller Access-Control System (TACACS, usually pronounced like tack-axe) refers to a family of related protocols handling remote authentication and related services for networked access control through a centralized server. The original TACACS protocol, which dates back to 1984, was used for communicating with an authentication server, common in older UNIX networks. Extended TACACS (XTACACS) is a proprietary extension to TACACS introduced by Cisco Systems in 1990 without backwards compatibility to the original protocol. Terminal Access Controller Access-Control System Plus (TACACS+) is a protocol developed by Cisco and released as an open standard beginning in 1993.

Decentralized Access Control

- A decentralized access control administration method gives control of access to the people closer to the resources
- In this approach, it is often the functional manager who assigns access control rights to employees.
- Changes can happen faster through this type of administration because not just one entity is making changes for the whole organization.
- There is a possibility for conflicts to arise that may not benefit the organization as because different managers and departments can practice security and access control in different ways.
- There is a possibility of certain controls to overlap, in which case actions may not be properly proscribed or restricted.
- This type of administration does not provide methods for consistent control, as a centralized method would.

A **Security Audit** is essentially an assessment of how effectively the organization's security policy is being implemented. It is an independent review and examination of an IT system's policy, records, and activities.

Information systems audit is important because it gives assurance that the IT systems are adequately protected, provide reliable information to users, and are properly managed to achieve their intended benefits. It also reduces the risk data tampering, data loss or leakage, service disruption and poor management of IT systems.

As Security Auditing and Testing (SAT) helps an organization to understand the state of security for internal reasons and provides assurance to external parties, it requires an attention of the highest degree. It helps to identify the gaps in the existing defenses.

Establishing audit objectives

After planning an Audit and before proceeding to perform the audit, one should establish the audit objectives. Following is a list of objectives the auditor should review:

- Personnel procedures and responsibilities including systems and cross-functional training
- Change management processes are in place and followed by IT and management personnel. Change Management refers to the efficient and prompt handling of all changes to control IT infrastructure, in order to minimize the number and impact of any related incidents upon service.
- Appropriate back up procedures are in place to minimize downtime and prevent loss of important data
- The data center has adequate physical security controls to prevent unauthorized access to the data center
- Adequate environmental controls are in place to ensure equipment is protected from fire and flooding.

Audit planning & preparation

The auditor should be adequately educated about the company and its critical business activities before conducting a data center review. The auditor should perform the following before conducting the review:

- Meet with IT management to determine possible areas of concern
- Review the current IT organization chart
- Review job descriptions of data center employees
- Research all operating systems, software applications and data center equipment operating within the data center
- Review the company's IT policies and procedures
- Evaluate the company's IT budget and systems planning documentation
- Review the data center's disaster recovery plan.

Performing an Audit

There is no standard security-audit process, but auditors typically accomplish their job through personal interviews, vulnerability scans, examination of OS and security-application settings, and network analyses, as well as by studying historical data such as event logs. Auditors also focus on the business's security policies to determine what they cover, how they are used and whether they are effective at meeting ongoing and future threats.

Generally, computer security audits are performed by:

- 1 Federal or State Regulators.
- 2 Corporate Internal Auditors.
- 3 External Auditors - Specialized in the areas related to technology auditing.
- 4 Consultants - Outsourcing the technology auditing where the organization lacks the specialized skill set.

First, the audit's scope should be decided and include all company assets related to information security, including computer equipment, phones, network, email, data and any access-related items, such as cards, tokens and passwords. Then, past and potential future asset threats must be reviewed. Anyone in the information security field should stay apprised of new trends, as well as security measures taken by other companies. Next, the auditing team should estimate the amount of destruction that could transpire under threatening conditions. There should be an established plan and controls for maintaining business operations after a threat has occurred, which is called an intrusion prevention system.

Performing the review

The next step is collecting evidence to satisfy data center audit objectives. This involves traveling to the data center location and observing processes and procedures performed within the data center. The following review procedures should be conducted to satisfy the pre-determined audit objectives:

- The auditor should observe and interview data center employees to satisfy their objectives.
- The auditor should verify that all data center equipment is working properly and effectively.
- All data center policies and procedures should be documented and located at the data center. Important documented procedures include: data center personnel job responsibilities, back up policies, security policies, employee termination policies, system operating procedures and an overview of operating systems.
- The auditor should assess the security of the client's data center with respect to physical security controls and environmental controls should be in place to ensure the security of data center equipment. These include: Air conditioning units, raised floors, humidifiers and uninterruptible power supply.
- Backup procedures - The auditor should verify that the client has backup procedures in place in the case of system failure. Clients may maintain a backup data center at a separate location that allows them to instantaneously continue operations in the instance of system failure.

Penetration Tests, Vulnerability Assessments And Security Audits

External Attackers often make use of already known vulnerabilities and exploits in order to infiltrate in systems and network. Taking appropriate defensive measures and adequate security design can mitigate this problem. This can also be achieved to a good extent by recognizing already existing exposed systems and their risks at regular intervals. The risks may be detected in the earlier stages and appropriate measures can be taken at an earlier stage.

Vulnerability assessment is a practice used to identify all potential vulnerabilities that could be exploited in an environment. The assessment can be used to evaluate physical security, personnel, or system and network

security. Vulnerability identification tools may be used to identify them. A list of every computer system by IP address and their associated vulnerabilities and steps on how to "fix" the vulnerabilities should then be generated.

Penetration test is carried out with the motive of "breaking into the network" using the known vulnerabilities. From here, the aim is to gain administrator or root access on the most critical system in the network. This gives complete access to the network to tamper with or modify the systems and the data on the systems. A penetration test is carried out to emulate what a real hacker would do and it proves to the company that the organization can indeed be penetrated.

Penetration testing is also referred to as ethical hacking. In most cases, the security professional can look at reports from a vulnerability scanner and understand the level of risk the company is facing.

Audit Controls

Review of Application Controls - It is the identification of the risks of deployed technology and minimization of the company's exposure to such risks, by ensuring that the necessary controls and security are in place.

Review of General Computer Controls - It is done for providing a secure and stable environment for the application systems running on various platforms within the company.

system may be lost if errors are found in operational systems

Objectives Of Controls

- To make sure data entering the computer are correct
- Check clerical handling of data before it is input to a computer
- Provide means of detecting and tracing errors which occur due to bad data or bad program
- Ensure legal requirements are met
- To guard against frauds

Access Control and Auditing

Physical and logical security

There are multiple types of security, physical and logical. Physical security involves things like locks or biometrics. Logical security examples consist of software safeguards including access control and auditing, user account management, violation and security activity reports, and firewalls.

Access control is a system enabling authorities to control access to areas and resources in a given physical facility or computer-based information system. The Password for a computer or PIN of an ATM system are forms of access control. Using an access control mechanism is important when persons seek to secure confidential, important, or sensitive information and equipment.

Auditing an Access Control System is a way of tracking the occurrence of entrance or attempted entrance into a system. This is important because it shows how successful the access control system is, as well as who was denied access, and if they attempt entrance more than once, what is their intention?

Logical Audit is done to check the following

- Strengths and weaknesses of Access Violations
- Security Activity Reports
- Logging activity reports
- Efficiency of firewalls
- Reports on violations of security
- Reports of Attempts by unauthorized persons to hack the system etc.

Professional Ethics for Auditors: To gain trust in an objective audit, it is necessary to uphold a set of professional ethics. The professional ethics must be upheld by individual persons as well as by companies providing services in the field of Information Security Auditing. The professional ethics consist of the following principles:

Honesty and confidentiality: Honesty is the foundation of trust and forms the basis for the reliability of an assessment. Since sensitive business processes and information are often found to be dependent on information security, the confidentiality of the information obtained during an audit and the discreet handling of the results and findings of the IS audit are an important basis for such work. IS auditors are aware of the value of the information they receive and who owns it, and will not disclose this information without the corresponding permission unless they are legally or professionally required to do so.

Expert knowledge: IS auditors only accept those jobs for which they have the requisite knowledge and skills as well as the corresponding experience and use these when performing their task. They continuously improve their knowledge as well as the effectiveness and quality of their work.

Objectivity and thoroughness: An IS auditor must demonstrate the highest possible level of expert objectivity and thoroughness when collecting, evaluating, and passing on information on the activities or business processes audited. The evaluation of all relevant circumstances must be performed impartially and may not be influenced by the auditor's own interests or the interests of others.

Objective presentation: An IS auditor has the duty to report the results of the examination precisely and truthfully to his client. This includes the impartial and understandable presentation of the facts in the IS audit reports, the constructive evaluation of the facts determined, and specific recommendations for improving the safeguards and processes.

Verifications and reproducibility: The rational basis for reliable and comprehensible conclusions and results is the clear and consistent documentation of the actual facts.

This also includes that the IS audit team follows a documented and reproducible methodology to come to its conclusions.

Compliance audit: A compliance audit is a comprehensive review of an organization's adherence to regulatory guidelines. Independent accounting, security or IT consultants evaluate the strength and thoroughness of compliance preparations. Auditors review security policies, user access controls and risk management procedures over the course of a compliance audit.

What is examined in a compliance audit will vary depending upon whether an organization is a public or private company, what kind of data it handles and if it transmits or stores sensitive financial data.

Information Security Policies: Organizations are giving more priority to development of information security policies, as protecting their assets is one of the prominent things that needs to be considered. Lack of clarity in information security policies can lead to severe damages which cannot be recovered. So an organization makes different strategies in implementing a security policy successfully. An information security policy provides management direction and support for information security across the organization.

Information in an organization needs to be secured properly against the consequences of breaches of confidentiality, integrity and availability. Proper security measures need to be implemented to control and secure information from unauthorized changes, deletions and disclosures. To find the level of security measures that need to be applied, a risk assessment is mandatory.

Security policies are intended to define what is expected from employees within an organization with respect to information systems. The objective is to guide or control the use of systems to reduce the risk to information assets. It also gives the staff who are dealing with information systems an acceptable use policy, explaining what is allowed and what not. Security policies of all companies are not same, but the key motive behind them is to protect assets. Security policies are designed with specific goals.

Information Security Audit Tools: Information Security Audit Tools include utilities and power tools, both open source and commercial.

Utility Tools : These are single-purpose tools that may either be native to the operating system or freely available. Utility tools require a manual approach, though they are often included in customized scripts--or even commercial products. You may even include native utilities, such as ping available on most platforms, used to determine if a network target responds to ICMP packets.

Pros: Utility tools are freely available and are tightly focused for a specific task, making them more efficient. They help in discovering vulnerabilities much faster than those found manually.

Cons: It requires skill to use them. For a large audit, manual testing is time-consuming and may produce inconsistent results, depending on the skill of the auditor.

Traceroute: A network tracing utility used to determine the network route to a host.

nslookup: used to determine domain ownership.

And open-source scripts, including:

Nmap: Free port-scanning utility.

Crack: Popular password-cracking tool used to determine if passwords are weak by attempting to break them.

John the Ripper: A password-cracking tool used primarily to discover Unix passwords.

binfo.c: A BIND version checker, binfo is a quick little script to pull back the version of named running on a remote name server.

ghba.c: A handy tool for extracting all the machine names and IP addresses of a given class B or C subnet.

Power Tools

Power Tools are multi-function bundled utilities to streamline and automate parts of the audit process. While some are open-source packages, many are commercial products with custom vulnerability databases.

Pros: Automated tools scan for vulnerabilities against a database. Alerts may be tied into help desk monitoring tools. In some cases, a scanning tool may be integrated with a firewall or intrusion detection management station. Some commercial scanners produce excellent reports detailing exposures and associated risk.

Cons: Scanners only check for vulnerabilities in their database, which must be current. Many scanners are marketed on the number of vulnerability checks performed. This isn't always a good indication of the tool's effectiveness. Often, vulnerabilities are misdiagnosed. A scanner can't accurately assess risk.

Some of the Open-source power tools are Nessus, SARA(The Security Auditor's Research Assistant), Whisker, etc. among many others.

Some of the commercial scanners available today are Internet Security System's Internet Scanner, eEye Digital Security's Retina, BindView's BV-Control, CORE Security Technology's Auditing Tools Suite and Foundstone's FoundScan.

Google

A real hacker thinks outside the box and learns to use tools in a way they may not have been intended. While the Google search engine is not, strictly speaking, an auditing tool, it's great for gathering information about a site. For example, trying entering "@DGET.com" (where "DGET" is your domain). Sometimes, this can yield some good data, such as a system administrator posting technical details about his site, which conveniently contains his account name. Google is like the Unix "grep" command on steroids.

Communicating Results

A final item to be considered is how to communicate with auditees, ie. the persons whose assets are being audited. When informing auditees of continuous audit activity results, it is important for the exchange to be independent and consistent.

Reporting to senior management on defined parameters

A typical audit report to the management and the management's response may look like the one shown in Table 1 below, but there are many other formats of the reports and Responses in use.

Table 1

Sl. No,	Findings	Impacts	Recommendations	Management Action / Response Plan	TimeLimit
1	The organization uses templates to configure new windows firewalls and servers, but does not have a documented technology configuration standard for other technologies such as DBMS, UNIX and LINUX operating systems etc.	The situation increases the risk of the unauthorized access to the organization's systems	The organization should continue documenting configuration standard based on the technologies in place.	Person Responsible : The chief information officer (CIO):The auditor's recommendations will be implemented. The security team will continue documenting configuration standards especially for UNIX and LINUX O.S.	No further action required
2	Patches are not up to date on LINUX based servers.	The situation increases the risk of the unauthorized access to the organization's systems. It also increases the risk of system failure.	Install the latest patches on the servers running on LINUX operating systems.	Person Responsible : The chief information Office. This was a result of complications in red hat LINUX maintenance contracts between the suppliers and the security team. The issue was settled in october 2014 and patches were installed.	No further action required.
3	We noted that users are their own workstation administrators, so that they can deactivate their workstation antivirus and idle system configurations.	This situation increases the risk of the unauthorized access to the organization's systems. It also increases the risk of system failure	Remove workstation administration privileges.	Person Responsible : The chief information officer. The situation was especially prevalent with windows 2000 with the latest servers users only have the privileges to do their job. There are a few exceptions. The situation will be remediated as we are upgrading all our systems.	November 1st 2014

Privacy Protection and IT Act

Objectives: At the end of this lesson you shall be able to

- describe information privacy
- describe the method of protecting privacy in information systems
- describe the best online privacy practices
- explain about what is IT Act
- describe about cyber crimes.

Introduction

Privacy is the ability of an individual or group to seclude their systems, data or information and share it selectively. When something is private to a person, it usually means that it is something special or sensitive.

Information or data privacy refers to the evolving relationship between technology and the legal right to, or public expectation of, privacy in the collection and sharing of data about one's self. Privacy concerns exist wherever uniquely identifiable data relating to a person or persons are collected and stored, in digital form or otherwise. In some cases these concerns refer to how data are collected, stored, and associated. In other cases the issue is who is given access to information. Other issues include whether an individual has any ownership rights to data about them, and/or the right to view, verify, and challenge that information.

Information security keeps unauthorized persons or systems from gaining access to restricted information. Privacy is the collection of rules and obligations that determine how and when access is to be authorized, in any medium. It follows that good security and privacy practices depend on one another. The domain of privacy partially overlaps security, including for instance the concepts of appropriate use, as well as protection of information.

The relationship between computer security and privacy lies in the fact that adequate computer security, or lack of it, is a determinant of the level of privacy that a computer user can expect. People use computers to perform many tasks, including business, banking, socializing and storing of private information. If there is a breach of computer security, it will have a negative effect on the way these types of tasks are carried out.

In the area of e-Commerce, the issue of computer security and privacy will determine the level of trust between the business parties. If there is any suspicion of a breach of security on either side, this will lead to a destruction of trust and an end to the business relationship. This includes risks and threats from third parties not even related to the business partners.

Improper or non-existent disclosure control can be the root cause for privacy issues. Data privacy issues can arise in response to information from a wide range of sources, such as:

- Health care records
- Investigations and proceedings
- Financial institutions and transactions
- Residence and geographic records
- Defence data
- Privacy breach
- Location-based service and geolocation
- Scientific research etc.

The challenge in data privacy is to share data while protecting personally identifiable information. The fields of data security and information security design and utilize software, hardware and human resources to address this issue. As the laws and regulations related to Data Protection are constantly changing, it is important to keep abreast of any changes in the law and continually reassess your compliance with data privacy and security regulations.

Protecting privacy in information systems: As a variety of information systems with differing privacy rules are interconnected and information is shared, policy appliances will be required to reconcile, enforce and monitor an increasing amount of privacy policy rules (and laws). There are two categories of technology to address privacy protection in commercial IT systems: communication and enforcement.

Policy Communication P3P : This is the Platform for Privacy Preferences. P3P is a standard for communicating privacy practices and comparing them to the preferences of individuals.

Policy Enforcement

- XACML - The Extensible Access Control Markup Language together with its Privacy Profile is a standard for expressing privacy policies in a machine-readable language which a software system can use to enforce the policy in enterprise IT systems.
- EPAL - The Enterprise Privacy Authorization Language is very similar to XACML, but is not yet a standard.
- WS-Privacy - "Web Service Privacy" will be a specification for communicating privacy policy in web services. For example, it may specify how privacy policy information can be embedded in the SOAP envelope of a web service message.

Protecting Privacy on the Internet

On the internet you almost always give away a lot of information about yourself. Unencrypted e-mails can be read by the administrators of the e-mail server where the connection is not encrypted (no https). Also the internet service provider and other parties sniffing the traffic of that connection are able to know the contents. Furthermore, the same applies to any kind of traffic generated on the internet (web browsing, instant messaging, ...) In order not to give away too much personal information, e-mails can be encrypted and browsing of webpages as well as other online activities can be done traceless via anonymizers, or, in cases those are not trusted, by open source distributed anonymizers, so called mix nets. Renowned open-source mix nets are I2P - The Anonymous Network or tor.

Protect Your Privacy

The following is a list of tips and guidelines to safeguard your privacy, personal information online and prevent fraud and abuse while using the Internet.

- **Get New Passwords:** Use different, strong passwords for each of your online accounts so if one is compromised the rest are safe. Strong passwords contains letters, numbers, different cases, and symbols. Check your password's strength here.
- **Close Old Online Accounts:** Unused online accounts are a liability. Hackers could use them to infiltrate your more important accounts . Get rid of them.
- **Reduce Your Friends List.**
- **Go Paperless:** Do not keep sensitive data online or in your mail accounts.
- **Shred Sensitive Documents:** Get rid of unwanted documents containing sensitive data. Dispose them securely, using a shredder.

Browser Privacy

Modern browsers have an impressive array of privacy enhancing capabilities and options. They can, for example, warn you before you visit suspicious or fraudulent websites and can also allow you to surf the web without downloading tracking files like cookies to your computer. Also, most browsers can inform you when a website uses SSL, a security measure that encrypts your data. When a website uses SSL a browser may indicate this to you by displaying a padlock icon (typically located on the bottom bar of your browser) or by highlighting the website's name in the address bar in green.

Visit only trusted websites. Use applications like Site Advisor etc. to know about the site you are opening. While websites today share more information, they also provide their users with great specificity and control over these sharing activities. On many websites you'll find that you can define your audience when you share personal information or content, whether it's an audience of one or the entire public.

Email has remained largely unchanged in the last decade. Methods of exploiting email, however, have evolved significantly and protecting personal information in email environments has become more challenging. In the past decade hacking has become more effective and phishing techniques, more elaborate. Here are some strategies for protecting your privacy when using email:

- 1 Use a secondary, "spam" email address
- 2 Use email service providers with strong security and spam filters
- 3 Exercise caution when opening emails
- 4 Recognize that email is evolving towards openness and interconnectivity
- 5 Use strong passwords and remember to sign-out

Best Online Privacy Practices

- 1 Minimize personal information sharing
- 2 Look for trustmarks on websites and verify their authenticity
- 3 Consider temporary credit card numbers when shopping online
- 4 Use strong passwords and remember to sign-out
- 5 Change your passwords frequently.
- 6 Use anti-virus and anti-spyware protection
- 7 Take advantage of browser privacy enhancing capabilities and options
- 8 Update your Browser and other tools.

Mobile Privacy

- 1 On mobile devices your personal information is more likely to be compromised via device theft or loss - take appropriate precautions
- 2 Your mobile device may be aware of your location and may share that data with applications and advertisers

CYBER CRIME ACT

In the era of cyber world as the usage of computers became more popular, there was expansion in the growth of technology as well, and the term '**Cyber**' became more familiar to the people. The evolution of Information Technology (IT) gave birth to the cyber space wherein internet provides equal opportunities to all the people to access any information, data storage, analyse etc. with the use of high technology. Due to increase in the number of netizens, misuse of technology in the cyberspace was clutching up which gave birth to cyber crimes at the domestic and international level as well.

Though the word Crime carries its general meaning as "a legal wrong that can be followed by criminal proceedings which may result into punishment" whereas **Cyber Crime** may be "unlawful acts wherein the computer is either a tool or target or both".

The world 1st computer specific law was enacted in the year 1970 by the German State of Hesse in the form of 'Data Protection Act, 1970' with the advancement of cyber technology. With the emergence of technology the misuse of technology has also expanded to its optimum level and then there arises a need of strict statutory laws to regulate the criminal activities in the cyber world and to protect technological advancement system. It is under these circumstances Indian parliament passed its "**INFORMATION TECHNOLOGY ACT, 2000**" on 17th oct to have its exhaustive law to deal with the technology in the field of e-commerce, e-governance, e-banking as well as penalties and punishments in the field of cyber crimes.

Cyber Crimes Actually

It could be hackers vandalizing your site, viewing confidential information, stealing trade secrets or intellectual property with the use of internet. It can also include 'denial of services' and viruses attacks preventing regular traffic from reaching your site. Cyber crimes are not limited to outsiders except in case of viruses and with respect to security related cyber crimes that usually done by the employees of particular company who can easily access the password and data storage of the company for their benefits. Cyber crimes also includes criminal activities done with the use of computers which further perpetuates crimes i.e. financial crimes, sale of illegal articles, pornography, online gambling, intellectual property crime, e-mail, spoofing, forgery, cyber defamation, cyber stalking, unauthorized access to Computer system, theft of information contained in the electronic form, e-mail bombing, physically damaging the computer system etc.

Classifications Of Cyber Crimes: Cyber Crimes which are growing day by day, it is very difficult to find out what is actually a cyber crime and what is the conventional crime so to come out of this confusion, cyber crimes can be classified under different categories which are as follows:

Cyber Crimes against Persons:

There are certain offences which affects the personality of individuals can be defined as:

Harassment via E-Mails: It is very common type of harassment through sending letters, attachments of files & folders i.e. via e-mails. At present harassment is common as usage of social sites i.e. Facebook, Twitter etc. increasing day by day.

Cyber-Stalking

It means expressed or implied a physical threat that creates fear through the use to computer technology such as internet, e-mail, phones, text messages, webcam, websites or videos.

Dissemination of Obscene Material

It includes Indecent exposure/Pornography (basically child pornography), hosting of web site containing these prohibited materials. These obscene matters may cause harm to the mind of the adolescent and tend to deprave or corrupt their mind.

Defamation

It is an act of imputing any person with intent to lower down the dignity of the person by hacking his mail account and sending some mails with using vulgar language to unknown persons mail account.

Hacking

It means unauthorized control/access over computer system and act of hacking completely destroys the whole data as well as computer programmes. Hackers usually hacks telecommunication and mobile network.

Cracking

It is amongst the gravest cyber crimes known till date. It is a dreadful feeling to know that a stranger has broken into your computer systems without your knowledge and consent and has tampered with precious confidential data and information.

E-Mail Spoofing

A spoofed e-mail may be said to be one, which misrepresents its origin. It shows it's origin to be different from which actually it originates.

SMS Spoofing:

Spoofing is a blocking through spam which means the unwanted uninvited messages. Here a offender steals identity of another in the form of mobile phone number and sending SMS via internet and receiver gets the SMS from the mobile phone number of the victim. It is very serious cyber crime against any individual.

Carding

It means false ATM cards i.e. Debit and Credit cards used by criminals for their monetary benefits through withdrawing money from the victim's bank account mala-fidely. There is always unauthorized use of ATM cards in this type of cyber crimes.

Cheating & Fraud

It means the person who is doing the act of cyber crime i.e. stealing password and data storage has done it with having guilty mind which leads to fraud and cheating.

Child Pornography

It involves the use of computer networks to create, distribute, or access materials that sexually exploit underage children.

Assault by Threat

Assault by Threat refers to threatening a person with fear for their lives or lives of their families through the use of a computer network i.e. E-mail, videos or phones.

Crimes Against Persons Property:

As there is rapid growth in the international trade where businesses and consumers are increasingly using computers to create, transmit and to store information in the electronic form instead of traditional paper documents.

There are certain offences which affects persons property which are as follows:

Intellectual Property Crimes

Intellectual property consists of a bundle of rights. Any unlawful act by which the owner is deprived completely or partially of his rights is an offence. The common form of IPR violation may be said to be software piracy, infringement of copyright, trademark, patents, designs and service mark violation, theft of computer source code, etc.

Cyber Squatting

It means where two persons claim for the same Domain Name either by claiming that they had registered the name first on by right of using it before the other or using something similar to that previously.

Cyber Vandalism

Vandalism means deliberately destroying or damaging property of another. Thus cyber vandalism means destroying or damaging the data when a network service is stopped or disrupted. It may include within its purview any kind of physical harm done to the computer of any person. These acts may take the form of the theft of a computer, some part of a computer or a peripheral attached to the computer.

Hacking Computer System: Hacktivism attacks those included Famous Twitter, blogging platform by unauthorized access/control over the computer. Due to the hacking activity there will be loss of data as well as computer. Also research especially indicates that those attacks were not mainly intended for financial gain too and to diminish the reputation of particular person or company.

Transmitting Virus: Viruses are programs that attach themselves to a computer or a file and then circulate themselves to other files and to other computers on a network. They usually affect the data on a computer, either by altering or deleting it. Worm attacks plays major role in affecting the computerize system of the individuals.

Cyber Trespass: It means to access someone's computer without the right authorization of the owner and does not disturb, alter, misuse, or damage data or system by using wireless internet connection.

Internet Time Thefts: Basically, Internet time theft comes under hacking. It is the use by an unauthorised person, of the Internet hours paid for by another person. The person who gets access to someone else's ISP user ID and password, either by hacking or by gaining access to it by illegal means, uses it to access the Internet without the other person's knowledge. You can identify time theft if the Internet time has to be recharged often, despite infrequent usage.

Cybercrimes Against Government:

There are certain offences done by group of persons intending to threaten the international governments by using internet facilities. It includes:

Cyber Terrorism

Cyber terrorism is a major burning issue in the domestic as well as global concern. The common form of these terrorist attacks on the Internet is by distributed denial of service attacks, hate websites and hate e-mails, attacks on sensitive computer networks etc. Cyber terrorism activities endanger the sovereignty and integrity of the nation.

Cyber Warfare

It refers to politically motivated hacking to conduct sabotage and espionage. It is a form of information warfare sometimes seen as analogous to conventional warfare although this analogy is controversial for both its accuracy and its political motivation.

Distribution of pirated software

It means distributing pirated software from one computer to another intending to destroy the data and official records of the government.

Possession of Unauthorized Information:

It is very easy to access any information by the terrorists with the aid of internet and to possess that information for political, religious, social, ideological objectives.

Cybercrimes Against Society at large:

An unlawful act done with the intention of causing harm to the cyberspace will affect large number of persons. These offences includes:

Child Pornography: It involves the use of computer networks to create, distribute, or access materials that sexually exploit underage children. It also includes activities concerning indecent exposure and obscenity.

Cyber Trafficking: It may be trafficking in drugs, human beings, arms weapons etc. which affects large number of persons. Trafficking in the cyberspace is also a gravest crime.

Online Gambling

Online fraud and cheating is one of the most lucrative businesses that are growing today in the cyber space. There are many cases that have come to light are those pertaining to credit card crimes, contractual crimes, offering jobs, etc.

Financial Crimes

This type of offence is common as there is rapid growth in the users of networking sites and phone networking where culprit will try to attack by sending bogus mails or messages through internet. Ex: Using credit cards by obtaining password illegally.

Forgery

It means to deceive large number of persons by sending threatening mails as online business transactions are becoming the habitual need of today's life style.

Affects To Whom

Cyber Crimes always affects the companies of any size because almost all the companies gain an online presence and take advantage of the rapid gains in the technology but greater attention to be given to its security risks. In the modern cyber world cyber crimes is the major issue which is affecting individual as well as society at large too.

Need of Cyber Law

Information technology has spread throughout the world. The computer is used in each and every sector wherein cyberspace provides equal opportunities to all for economic growth and human development. As the user of cyberspace grows increasingly diverse and the range of online interaction expands, there is expansion in the cyber crimes i.e. breach of online contracts, perpetration of online torts and crimes etc. Due to these consequences there was need to adopt a strict law by the cyber space authority to regulate criminal activities relating to cyber and to provide better administration of justice to the victim of cyber crime. In the modern cyber technology world it is very much necessary to regulate cyber crimes and most importantly cyber law should be made stricter in the case of cyber terrorism and hackers

Penalty For Damage To Computer System

According to the Section: 43 of 'Information Technology Act, 2000' whoever does any act of destroys, deletes, alters and disrupts or causes disruption of any computer with the intention of damaging of the whole data of the computer system without the permission of the owner of the computer, shall be liable to pay fine upto 1crore to the person so affected by way of remedy. According to the Section:43A which is inserted by 'Information Technology (Amendment) Act, 2008' where a body corporate is maintaining and protecting the data of the persons as provided by the central government, if there is any negligent act or failure in protecting the data/ information then a body corporate shall be liable to pay compensation to person so affected. And Section 66 deals with 'hacking with computer system' and provides for imprisonment up to 3 years or fine, which may extend up to 2 years or both.

Working with Cloud Services

Objectives: At the end of this lesson you shall be able to

- introduction to Cloud computing
- types of cloud computing
- disadvantages and benefits of cloudcomputing.

Introduction to cloud computing

Cloud Computing: Benefits, Disadvantages & Types of Cloud Computing Services

Cloud computing is one of the hottest catchphrases in business today. It has transformed the way organizations store, access and share information, collaborate and manage computing resources. With the advent of the internet, cloud computing has provided new ways of conducting business by allowing companies to rise above the conventional on-premises IT infrastructure.

Cloud computing offers modern businesses flexibility, efficiency, scalability, security, increased collaboration and reduced costs. While the COVID-19 pandemic has accelerated cloud adoption, the reliance on cloud technologies is set to continue in 2022, especially with hybrid work taking center stage. So, whether an organization already uses cloud services or is planning to in the coming year, it is imperative to understand the basics of cloud computing in order to take full advantage of cloud-powered solutions.

In this blog, we will explore what exactly cloud computing is, how it works, its benefits and disadvantages, and how companies can protect their SaaS data better.



What is cloud computing?

According to ZDNet, "cloud computing is the delivery of on-demand computing services - from applications to storage and processing power - typically over the internet and on a pay-as-you-go basis."

In simplest terms, the cloud refers to the internet. When organizations store data in virtual data centers or access programs using an internet connection instead of relying on their device's hard drive or on-premises IT infrastructure, it means they are operating in the cloud.

Cloud computing can be as simple as "servers in a third-party data center" or entire serverless workloads that are infinitely scalable and geo-redundant. Cloud servers and services are scalable and elastic.

How does cloud computing work?

Cloud computing is the delivery of computing resources, such as IT infrastructure or data center over the internet. This model allows businesses to rent storage space or access software programs from a cloud service provider, instead of building and maintaining their own IT infrastructure or data center. One major benefit of using cloud computing services is that companies pay only for the resources they use.

To better understand its technical aspects, cloud computing processes can be divided into frontend and backend. The frontend component allows users to access data and programs stored in the cloud through an internet browser or by using a cloud computing application. The backend consists of servers, computers and databases that store the data.

History of cloud computing

According to Technology Review, the phrase "cloud computing" was first mentioned in 1996 in a Compaq internal document.

The year 1999 was a milestone for cloud computing when Salesforce became the first company to deliver enterprise applications over the internet. This was also the beginning of Software-as-a-Service (SaaS).

In 2002, Amazon launched Amazon Web Services (AWS), which was another significant development in cloud computing. Its suite of cloud-based services included storage, computation and even human intelligence. In 2006, Amazon launched Elastic Compute Cloud (EC2), allowing businesses as well as individuals to rent virtual computers and run their own computer applications.

The year 2009 saw yet another giant milestone in cloud computing as Google Workspace (now Google Workspace) started to provide browser-based enterprise applications. In the same year, Microsoft entered the cloud computing arena with Microsoft Azure, and soon companies like Oracle and HP followed suit.

What are examples of cloud computing?

Cloud computing includes everything from virtual machines to databases to entire serverless applications. Some examples of cloud computing include:

Salesforce: Salesforce.com is a SaaS provider that specializes in customer relationship management (CRM). The company provides enterprise applications to help align marketing, sales, customer services, etc., and allows users to work from anywhere.

Digital Ocean: This company is a New York-based Infrastructure-as-a-Service (IaaS) provider for software developers. Businesses use DigitalOcean to deploy and scale applications that run simultaneously across multiple cloud servers.

Microsoft Azure: Microsoft Azure is a fine example of a Platform-as-a-Service (PaaS) that supports the entire application development lifecycle, right from development to deployment and beyond. Azure provides a plethora of tools, languages and frameworks to developers.

Dropbox: Dropbox is a cloud-based file hosting service that allows users to store and sync files to their devices so they can access them from anywhere. It also allows users to share large files, including images and videos via the internet, facilitating effective collaboration.

What is the importance of cloud computing?

Before cloud computing came into existence, companies were required to download applications or programs on their physical PCs or on-premises servers to be able to use them. For any organization, building and managing its own IT infrastructure or data centers is a huge challenge. Even for those who own their own data centers, allocating a large number of IT administrators and resources is a struggle.

The introduction of cloud computing and virtualization was a paradigm shift in the history of the technology industry. Rather than creating and managing their own IT infrastructure and paying for servers, power and real estate, etc., cloud computing allows businesses to rent computing resources from cloud service providers. This helps businesses avoid paying heavy upfront costs and the complexity of managing their own data centers. By renting cloud services, companies pay only for what they use such as computing resources and disk space. This allows companies to anticipate costs with greater accuracy.

Since cloud service providers do the heavy lifting of managing and maintaining the IT infrastructure, it saves a lot of time, effort and money for businesses. The cloud also gives organizations the ability to seamlessly upscale or downscale their computing infrastructure as and when needed. Compared to the traditional on-premises data center model, the cloud offers easy access to data from anywhere and on any device with internet connectivity, thereby enabling effective collaboration and enhanced productivity.

What are the most common uses of cloud computing?

From startups to large corporations and government agencies, every organization uses the cloud to access technology services to streamline workflows, improve communication, productivity, service delivery and more. Listed below are some of the most common uses of cloud computing.

- **Storage:** One of the most common uses of cloud computing is file storage. While there are several options to store and access data, such as hard drives on PCs, external hard drives, USB drives, etc., cloud

storage enables businesses to seamlessly access data from anywhere and on any device with an internet connection. Cloud storage services like Amazon S3, DropBox or OneDrive provide secure access to data and also allows businesses to upscale and downscale storage space based on their requirements.

- **Database:** Cloud database is another popular business use case. IBM defines cloud database as "a database service built and accessed through a cloud platform." A cloud database delivers most of the same functionalities as a traditional database, but with additional benefits such as flexibility, cost savings, failover support, specialized expertise and more.
- **Web applications:** Web applications are a must-have tool for businesses today. Powered by cloud technology, anyone can access web-based apps using a web browser, providing instant remote access to information. This allows business professionals to communicate with customers and provide them with required information while they're on the go, and helps them collaborate with colleagues from anywhere.
- **Collaboration:** Due to its easy accessibility, integration, flexibility, security and ease of use, cloud-based tools, such as Microsoft 365 and Google Workspace, have become the obvious choice for businesses looking to collaborate both internally across departments and externally with clients. Gmail, Google docs, Microsoft Outlook, Microsoft Word, Teams, etc., are powerful business tools designed to enhance collaboration and productivity.
- **SaaS applications:** Software-as-a-Service (SaaS) applications, such as Salesforce, allow businesses to store, organize and maintain data, as well as automate marketing and manage clients efficiently. SaaS solutions are highly functional and do not require software and/or hardware management.

What are the different types of cloud computing?

There are four main types of cloud computing: public, private, hybrid and multicloud.

Public cloud

VMware defines public cloud as "an IT model where on-demand computing services and infrastructure are managed by a third-party provider and shared with multiple organizations using the public internet." Cloud service providers offer various services like Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS) and SaaS to individuals and businesses who rent these services on a monthly or pay-per-use basis. Amazon Web Services (AWS), Microsoft Azure, Google Cloud, Alibaba Cloud and IBM Cloud are the top five cloud providers.

Private cloud

A private cloud or an internal cloud is where the IT infrastructure (hardware and software resources) is solely dedicated to a single organization, unlike a public cloud where the computing resources are shared among multiple tenants. A private cloud environment is ideal for businesses

for whom meeting regulatory requirements, security and control are a priority. Traditionally, a private cloud is hosted at a company's data center and uses its own hardware. However, an organization may outsource hosting to a third-party provider who remotely manages the computing resources.

Hybrid cloud

A hybrid cloud is a combination of both public cloud and private cloud environments. Businesses use this model to supplement their compute capacity. When the capacity of a private cloud reaches its peak, businesses can leverage public cloud to enhance the capabilities of the private cloud. Hybrid cloud enables businesses to scale compute capacity up or down depending on the traffic or service demands. This eliminates the need to purchase and maintain new servers, allowing businesses to save cost, time and effort.

Multicloud

Multicloud is the practice of using a combination of clouds - two or more public or private clouds, or a combination of both, from several cloud providers. A multicloud approach allows businesses to select the best services from different cloud vendors based on their budgets, technical requirements, geographic locations and so on. This model enables businesses to use different clouds for different purposes. For instance, an organization can use one cloud for software development and testing, another cloud for data backup and disaster recovery, and other for data analytics.

What are the three different types of cloud computing services?

The three types of cloud computing services are Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS) and Software-as-a-Service (SaaS).

Infrastructure-as-a-Service (IaaS)

IaaS is a cloud computing service where cloud providers deliver and manage virtualized computing infrastructure over the internet. Instead of creating an in-house IT infrastructure, businesses can access essential resources, such as operating systems, networking, storage space, development tools, etc., on demand. This saves hardware and software costs as well as minimizes the burden of IT staff.

Platform-as-a-Service (PaaS)

PaaS allows businesses to concentrate on the development, deployment and management of software applications and services without having to worry about the underlying infrastructure since cloud providers do the heavy lifting. With PaaS, developers and programmers gain access to not only IT infrastructure but also application/software platform and solution stack. Some of the examples of PaaS include AWS Elastic Beanstalk, Google App Engine and Microsoft Azure.

Software-as-a-Service (SaaS)

SaaS provides businesses with ready-to-use software that is delivered to users over the internet. All of the underlying infrastructure, including hardware, software, data storage, patch management and hardware/software updates, are managed by SaaS providers. SaaS is a subscription-based model, which requires businesses to subscribe to the services they want to use. Users can access SaaS applications directly through web browsers, which eliminates the need to download or install them. SaaS allows users to access web-based solutions from anywhere and at any time with an active internet connection. Some popular SaaS solutions include Microsoft 365, Google Workspace and Salesforce.

What are the benefits of cloud computing?

Cloud computing enables businesses to operate from virtually anywhere and with more efficiency. Some benefits of cloud computing include:

- **Cost savings:** One of the greatest benefits of cloud computing is reduced costs. Since businesses do not need to build their own IT infrastructure or purchase hardware or equipment, it helps companies reduce capital expenses significantly.
- **Flexibility/scalability:** Cloud computing offers greater flexibility to businesses of all sizes. Whether they require extra bandwidth, computing power or storage space, they can seamlessly scale up or down computing resources depending on their needs and budget.
- **Security:** Data security is a major concern for businesses today. Cloud vendors provide advanced security features like authentication, access management, data encryption, etc., to ensure sensitive data in the cloud is securely handled and stored.
- **Mobility:** Cloud computing allows users to access corporate data from any device, anywhere and at any time, using the internet. With information conveniently available, employees can remain productive even on the go.
- **Increased collaboration:** Cloud applications allow businesses to seamlessly communicate and securely access and share information, making collaboration simple and hassle-free. Cloud computing empowers multiple users to edit documents or work on files simultaneously and in a transparent manner.
- **Disaster recovery:** Data loss and downtime can cause irreparable damage to businesses of any size. Major cloud vendors are well-equipped to withstand unforeseen disruptive events, such as hardware/software failure, natural disasters and power outages, to ensure high application availability and business continuity.

- **Automatic updates:** Performing manual organization-wide software updates can take up a lot of valuable IT staff time. However, with cloud computing, service providers regularly refresh and update systems with the latest technology to provide businesses with up-to-date software versions, latest servers and upgraded processing power.

What are the disadvantages of cloud computing?

The advantages of operating in the cloud are immense. However, there are certain disadvantages that companies should be aware of before deciding to transition to the cloud. Listed below are the top five disadvantages of cloud computing.

- 1 Downtime:** Since cloud computing systems are completely reliant on the internet, without an active internet connection, businesses cannot access the data or applications hosted in the cloud. Google suffered three severe outages in 2020 that affected the majority of its services and users across the globe.
- 2 Vendor lock-in:** Migrating a company's workloads and services from one cloud provider to another is a major challenge in cloud computing. Differences between cloud environments may cause compatibility or integration issues. If the transition isn't handled properly, it could expose an organization's data to unnecessary security vulnerabilities.
- 3 Limited control:** Since the cloud infrastructure is wholly owned and managed by the cloud vendor, businesses using cloud computing services have limited control over their data, applications and services. Therefore, it's important to have a proper end-user license agreement (EULA) in place to understand what a business can do and can't do within a cloud infrastructure.
- 4 Security:** One of the major concerns of storing a company's sensitive data in the cloud is security. Although cloud service providers implement advanced security measures, storing confidential files on remote servers that are entirely owned and operated by a third party always opens up security risks. When an organization adopts a cloud computing model, the IT security responsibility is shared between the cloud vendor and the user. As such, each party is responsible for the assets, processes and functions they control.
- 5 Data loss or theft:** Storing crucial data in virtual data centers can open the doors to a variety of risks that could lead to data loss, such as cloud misconfiguration, information theft, security breach, stolen credentials, etc. Moreover, cloud service providers, such as Microsoft and Google, follow a shared responsibility model, where the vendor assumes responsibility for application availability and everything that entails, while the customer retains responsibility for application data, administration and user management.

Improve SaaS data protection with Spanning Backup

According to Statista, as of 2021, around 50% of all corporate data is stored in the cloud. The data suggests that businesses globally trust their cloud service providers with their sensitive data. Regardless of which cloud vendor a company chooses, they must be careful that the benefits of the cloud do not outweigh the underlying security risks.

Businesses using SaaS solutions such as Microsoft 365, Google Workspace and Salesforce lose data every day. Many companies tend to believe that SaaS vendors are responsible for protecting their data. However, that's not the case. While SaaS providers ensure application uptime and availability, data protection is the customers' responsibility.

As such, businesses need a reliable SaaS backup solution that can protect their valuable data against the most common causes of data loss like phishing, ransomware and malware attacks, human error, malicious behavior, and configuration and sync errors.

Spanning backup and end-to-end protection solutions for Microsoft 365, Google Workspace and Salesforce fill the gaps in native functionality to protect critical data from loss caused by these threats, reducing the risk of compromise and enabling end users and administrators to quickly find and restore data to its original state in just a few clicks.

Best Resources to Learn Cloud Computing

Today, many professionals are pursuing cloud computing as their career. To be successful in this career, they need to stay current with the latest news and trends to improve their skills and learn cloud computing. The following are some of the top resources to learn cloud computing and remain in touch with the latest trends in the industry:

Online Cloud Computing Courses

Online courses are one of the best ways to learn cloud computing. Many online courses offer videos, articles, eBooks, reports, hands-on labs, and assessments to help the learners. Some of the best cloud computing courses online are listed below:

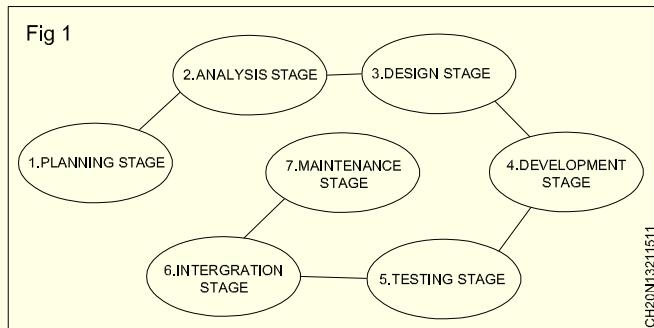
- Introduction to Cloud Computing on Amazon AWS for Beginners on Udemy
- Introduction to Cloud Computing on Coursera
- Cloud Computing with AWS Training on Internshala
- Cloud Computing Basics: Enhance your career as Cloud Engineer on Udemy
- Introduction to Cloud Computing by Microsoft on edX
- Cloud Computing by NPTEL
- Cloud Computing Basics (Cloud 101) on Coursera
- TOTAL: Cloud Computing / CompTIA Cloud+ Cert. (CV0-002) on Udemy

Identify Phases of the Application Development Life Cycle

Objectives: At the end of this lesson you shall be able to

- identify phases of the application development life cycle
- benefits of SDLC.

Identify phases of the application development life cycle (Fig 1)



System

The software development process is normally long and tedious. But project managers and system analysts can leverage software development life cycles to outline, design, develop, test, and eventually deploy information systems or software products with greater regularity, efficiency, and overall quality.

What is System Development Life Cycle?

A system development life cycle or SDLC is essentially a project management model. It defines different stages that are necessary to bring a project from its initial idea or conception all the way to deployment and later maintenance.

System Development Life Cycle US Guide (Fig 2)

In this guide, we'll break down everything you need to know about the system development life cycle, including all of its stages. We'll also go over the roles of system analysts and the benefits your project might see by adopting SDLC.

7 Stages of the System Development Life Cycle



There are seven primary stages of the modern system development life cycle. Here's a brief breakdown:

- Planning Stage

- Feasibility or Requirements of Analysis Stage
- Design and Prototyping Stage
- Software Development Stage
- Software Testing Stage
- Implementation and Integration
- Operations and Maintenance Stage

Now let's take a closer look at each stage individually.

Planning Stage

Before we even begin with the planning stage, the best tip we can give you is to take time and acquire proper understanding of app development life cycle.

The planning stage (also called the feasibility stage) is exactly what it sounds like: the phase in which developers will plan for the upcoming project.

It helps to define the problem and scope of any existing systems, as well as determine the objectives for their new systems.

By developing an effective outline for the upcoming development cycle, they'll theoretically catch problems before they affect development.

And help to secure the funding and resources they need to make their plan happen.

Perhaps most importantly, the planning stage sets the project schedule, which can be of key importance if development is for a commercial product that must be sent to market by a certain time.

Analysis Stage

The analysis stage includes gathering all the specific details required for a new system as well as determining the first ideas for prototypes.

Developers may:

- Define any prototype system requirements
- Evaluate alternatives to existing prototypes
- Perform research and analysis to determine the needs of end-users

Furthermore, developers will often create a software requirement specification or SRS document.

This includes all the specifications for software, hardware, and network requirements for the system they plan to build. This will prevent them from overdrawing funding or resources when working at the same place as other development teams.

Design Stage

The design stage is a necessary precursor to the main developer stage.

Developers will first outline the details for the overall application, alongside specific aspects, such as its:

- User interfaces
- System interfaces
- Network and network requirements
- Databases

They'll typically turn the SRS document they created into a more logical structure that can later be implemented in a programming language. Operation, training, and maintenance plans will all be drawn up so that developers know what they need to do throughout every stage of the cycle moving forward.

Once complete, development managers will prepare a design document to be referenced throughout the next phases of the SDLC.

Development Stage (Fig 3)

The development stage is the part where developers actually write code and build the application according to the earlier design documents and outlined specifications.

This is where Static Application Security Testing or SAST tools come into play.

Product program code is built per the design document specifications. In theory, all of the prior planning and outlined should make the actual development phase relatively straightforward.

Fig 3



Developers will follow any coding guidelines as defined by the organization and utilize different tools such as compilers, debuggers, and interpreters.

Programming languages can include staples such as C++, PHP, and more. Developers will choose the right programming code to use based on the project specifications and requirements.

Testing Stage (Fig 4)

Fig 4



Building software is not the end.

Now it must be tested to make sure that there aren't any bugs and that the end-user experience will not negatively be affected at any point.

During the testing stage, developers will go over their software with a fine-tooth comb, noting any bugs or defects that need to be tracked, fixed, and later retested.

It's important that the software overall ends up meeting the quality standards that were previously defined in the SRS document.

Depending on the skill of the developers, the complexity of the software, and the requirements for the end-user, testing can either be an extremely short phase or take a very long time. Take a look at our top 10 best practices for software testing projects for more information.

Implementation and Integration Stage

After testing, the overall design for the software will come together. Different modules or designs will be integrated into the primary source code through developer efforts, usually by leveraging training environments to detect further errors or defects.

The information system will be integrated into its environment and eventually installed. After passing this stage, the software is theoretically ready for market and may be provided to any end-users.

Maintenance Stage (Fig 5)

The SDLC doesn't end when software reaches the market. Developers must now move into a maintenance mode and begin practicing any activities required to handle issues reported by end-users.

Furthermore, developers are responsible for implementing any changes that the software might need after deployment.

This can include handling residual bugs that were not able to be patched before launch or resolving new issues that crop up due to user reports. Larger systems may require longer maintenance stages compared to smaller systems.

Fig 5



Role of System Analyst

An SDLC's system analyst is, in some ways, an overseer for the entire system. They should be totally aware of the system and all its moving parts and can help guide the project by giving appropriate directions.

The system analyst should be:

- An expert in any technical skills required for the project
- A good communicator to help command his or her team to success
- A good planner so that development tasks can be carried out on time at each phase of the development cycle

Thus, systems analysts should have an even mix of interpersonal, technical, management, and analytical skills altogether. They're versatile professionals that can make or break an SDLC.

Their responsibilities are quite diverse and important for the eventual success of a given project. Systems analysts will often be expected to:

- Gather facts and information
- Make command decisions about which bugs to prioritize or what features to cut
- Suggest alternative solutions
- Draw specifications that can be easily understood by both users and programmers
- Implement logical systems while keeping modularity for later integration
- Be able to evaluate and modify the resulting system as is required by project goals
- Help to plan out the requirements and goals of the project by defining and understanding user requirements

6 Basic SDLC Methodologies

Although the system development life cycle is a project management model in the broad sense, six more specific methodologies can be leveraged to achieve specific results or provide the greater SDLC with different attributes.

Waterfall Model (Fig 6)

The waterfall model is the oldest of all SDLC methodologies. It's linear and straightforward and requires development teams to finish one phase of the project completely before moving on to the next.

Each stage has a separate project plan and takes information from the previous stage to avoid similar issues (if encountered). However, it is vulnerable to early delays and can lead to big problems arising for development teams later down the road.

Fig 6



Iterative Model (Fig 7)

The iterative model focuses on repetition and repeat testing. New versions of a software project are produced at the end of each phase to catch potential errors and allow developers to constantly improve the end product by the time it is ready for market.

Fig 7



One of the upsides to this model is that developers can create a working version of the project relatively early in their development life cycle, so implement the changes are often less expensive.

Spiral Model

Spiral models are flexible compared to other methodologies. Projects pass through four main phases again and again in a metaphorically spiral motion.

It's advantageous for large projects since development teams can create very customized products and incorporate any received feedback relatively early in the life cycle.

V-Model

The V-model (which is short for verification and validation) is quite similar to the waterfall model. A testing phase is incorporated into each development stage to catch potential bugs and defects.

It's incredibly disciplined and requires a rigorous timeline. But in theory, it illuminates the shortcomings of the main waterfall model by preventing larger bugs from spiraling out of control.

Big Bang Model

The Big Bang model is incredibly flexible and doesn't follow a rigorous process or procedure. It even leaves detailed planning behind. It's mostly used to develop broad ideas when the customer or client isn't sure what they want. Developers simply start the project with money and resources.

Their output may be closer or farther from what the client eventually realizes they desire. It's mostly used for smaller projects and experimental life cycles designed to inform other projects in the same company. (Fig 8)

Fig 8



Agile Model

The agile model is relatively well-known, particularly in the software development industry.

The agile methodology prioritizes fast and ongoing release cycles, utilizing small but incremental changes between releases. This results in more iterations and many more tests compared to other models.

Theoretically, this model helps teams to address small issues as they arise rather than missing them until later, more complex stages of a project. (Fig 9)

Fig 9



Benefits of SDLC: SDLC provides a number of advantages to development teams that implement it correctly.

Clear Goal Descriptions: Developers clearly know the goals they need to meet and the deliverables they must achieve by a set timeline, lowering the risk of time and resources being wasted.

Proper Testing Before Installation: SDLC models implement checks and balances to ensure that all software is tested before being installed in greater source code.

Clear Stage Progression: Developers can't move on to the next age until the prior one is completed and signed off by a manager.

Member Flexibility: Since SDLCs have well-structured documents for project goals and methodologies, team members can leave and be replaced by new members relatively painlessly.

Perfection Is Achievable: All SDLC stages are meant to feed back into one another. SDLC models can therefore help projects to iterate and improve upon themselves over and over until essentially perfect.

No One Member Makes or Breaks the Project

Again, since SDLCs utilize extensive paperwork and guideline documents, it's a team effort and losing one even major member will not jeopardize the project timeline.

What You Need to Know About System Development Life Cycle

Where is SDLC Used?

System development life cycles are typically used when developing IT projects.

Software development managers will utilize SDLCs to outline various development stages, make sure everyone completes stages on time and in the correct order, and that the project is delivered as promptly and as bug-free as possible.

SDLCs can also be more specifically used by systems analysts as they develop and later implement a new information system.

What SDLC Model is Best?

It largely depends on what your team's goals and resource requirements are.

The majority of IT development teams utilize the agile methodology for their SDLC. However, others may prefer the iterative or spiral methodologies.

All three of these methods are popular since they allow for extensive iteration and bug testing before a product is integrated with greater source code or delivered to market.

DevOps methodologies are also popular choices. And if you ever need a refresher course on what is DevOps, you needn't worry as our team at CloudDefense has got you covered!

What Does SDLC Develop?

SDLC can be used to develop or engineer software, systems, and even information systems. It can also be used to develop hardware or a combination of both software and hardware at the same time.

FAQs

What Were the 5 Original Phases of System Development Life Cycle?

The systems development life cycle originally consisted of five stages instead of seven. These included planning, creating, developing, testing, and deploying. Note that it left out the major stages of analysis and maintenance.

What Are the 7 Phases of SDLC?

The new seven phases of SDLC include planning, analysis, design, development, testing, implementation, and maintenance.

What is System Development Life Cycle in MIS?

In the greater context of management information systems or MIS, SDLC helps managers to design, develop, test, and deploy information systems to meet target goals.

Conclusion

Ultimately, any development team in both the IT and other industries can benefit from implementing system development life cycles into their projects. Use the above guide to identify which methodology you want to use in conjunction with your SDLC for the best results.

Programming language (Python) Use Python from command line

Objectives: At the end of this lesson you shall be able to

- introduction to Python
- features of Python
- comments of Python
- variable.

Python is a general purpose, high-level, interpreted programming language developed by Guido van Rossum in the late 1980s at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is one of the most popular and widely used programming language used for set of tasks including console based, GUI based, web programming and data analysis.

Python is a easy to learn and simple programming language so even if you are new to programming, you can learn python without facing any problems.

Fact: Python is named after the comedy television show Monty Python's Flying Circus.

- Python Applications
- Installation of Python

Features of Python

Python provides lots of features that are listed below.

- Easy to Learn and Use

Python is easy to learn and use compared with other programming languages. It is developer-friendly and high level programming language.

- Interpreted Language

Python is an interpreted language because no need of compilation. This makes debugging easy and thus suitable for beginners.

- Cross-platform Language

Python can run equally on different platforms such as Windows, Linux, Unix and Macintosh etc. So, we can say that Python is a portable language.

- Free and Open Source

The Python interpreter is developed under an open-source license, making it free to install, use, and distribute.

- Object-Oriented Language

Python supports object oriented language and concepts of classes and objects come into existence.

- GUI Programming Support

Graphical user interfaces can be developed using Python.

- Integrated

It can be easily integrated with languages like C, C++, and JAVA etc.

Comments in Python

In general, Comments are used in a programming language to describe the program or to hide the some part of code from the interpreter.

Comments in Python can be used to explain any program code. It can also be used to hide the code as well.

Comment is not a part of the program, but it enhances the interactivity of the program and makes the program readable.

Python supports two types of comments:

- Single Line Comment
- Multi Line Comment

1 Single Line Comment

In case user wants to specify a single line comment, then comment must start with '#'

format:

```
# This is single line comment
```

Example: "scomment.py"

```
# This is single line comment.
print("Hello Python")
```

Output: \$python3 scomment.py

Hello Python

2 Multi Line Comment

Multi lined comment can be given inside triple quotes. The must start at begining of the line.

format:

```
'''This
is
Multiline comment'''
```

Example: "mcomment.py"

```
'''This
is
Multi line comment'''
print("Hello Python")
```

Output: \$python3 mcomment.py

Hello Python

Example :

Example: "comments.py"

This example demonstrates usage of Comments

... print() used to

print/display

text on screen ""

```
print("Welcome to Python")
```

#Assign value to variables

a=20

b=30

#print sum of two numbers

```
print("Sum is")
```

```
print(a+b)
```

Output: \$python3 comments.py

Welcome to Python

Sum is

50

Variables in Python

A variable is a named memory location in which we can store values for the particular program. In other words, Variable is a name which is used to refer memory location. Variable also known as identifier and used to hold value.

Creating Variables

In Python, We don't need to declare explicitly variable in Python. When we assign any value to the variable that variable is declared automatically.

In Python, We don't need to specify the type of variable because Python is a loosely typed language. I.e. In loosely typed language no need to specify the type of variable because the variable automatically changes its datatype based on assigned value.

For Example:

```
?? a=10????Where variable a is an integer datatype.
```

```
?? b="Glance"????Where variable b is string datatype.
```

Rules for naming variable:

- Variable names can be a group of both letters and digits, but they have to begin with a letter or an underscore.
- It is recommended to use lowercase letters for variable name. 'SUM' and 'sum' both are two different variables.

Example: "vardemo.py"

a=10 #integer

b="StudyGlance" #string

c=12.5 #float

print(a)

print(b)

print(c)

Output: \$python3 vardemo.py

10

StudyGlance

12.5

Assign Value to Multiple Variables

Python allows us to assign a value to multiple variables and multiple values to multiple variables in a single statement which is also known as multiple assignment.

Assign single value to multiple variables

Example: "vardemo1.py"

x=y=z=50

print x

print y

print z

Output: \$python3 vardemo1.py

50

50

50

Assign multiple values to multiple variables

Example: "vardemo2.py"

a,b,c=5,10,15

print a

print b

print c

Output: \$python3 vardemo2.py

5

10

15

Perform Operations using Data Types and Operators

Objectives: At the end of this lesson you shall be able to

- data types , Operators in Python.

In general, Data Types specifies what type of data will be stored in variables. Variables can hold values of different data types. Python is a dynamically typed or loosely typed language, hence we need not define the type of the variable while declaring it. The interpreter implicitly binds the value with its type.

I.e. In loosely typed language no need to specify the type of variable because the variable automatically changes its datatype based on assigned value.

Python provides the type() function which enables us to check the type of the variable.

Python provides following standard data types, those are

1 Numbers

2 String

1 Numbers

Number stores numeric values. Python creates Number type variable when a number is assigned to a variable.

There are three numeric types in Python

a int

b float

c complex

a int

Int, or integer, is a whole number, positive or negative, without decimals, of unlimited length.

format:

a=10

b=-12

c=123456789

b float

Float or "floating point number" is a number, positive or negative, containing one or more decimals.

format:

X=1.0

Y=12.3

Z=-13.4

c complex

Complex numbers are written with a "j" as the imaginary part.

format:

A =2+5j

B =-3+4j

C =-6j

Example: "datatypedemo1.py"

a =10

b =10.5

c =2.14j

print("Datatype of Variable a : ",type(a))

print("Datatype of Variable b : ",type(b))

print("Datatype of Variable c : ",type(c))

Output: \$python3 datatypedemo1.py

Datatype of Variable a :<class 'int'>

Datatype of Variable b :<class 'float'>

Datatype of Variable c :<class 'complex'>

2 String

The string can be defined as the sequence of characters represented in the quotation marks. In python, we can use single, double, or triple quotes to define a string.

In the case of string handling, the operator + is used to concatenate two strings as the operation "hello"+ " python" returns "hello python".

format:

S1='Welcome'

S2="to Python"

S3="""world"""

Example:

a ="Welcome" #using double quotes

b ='Python' #using single quotes

c ="""World"""\n#using triple quotes

print("Datatype of Variable a : ",type(a))

print(a+b)#to concatenate two strings

Output:

Datatype of Variable a :<class 'str'>

Welcome Python

Operators in Python

The operator can be defined as a symbol which is responsible for a particular operation between two operands.

Python provides a variety of operators described as follows.

- Arithmetic operators
- Assignment Operators
- Comparison operators
- Bitwise Operators
- Identity Operators
- Logical Operators
- Membership Operators

1 Arithmetic operators

Arithmetic operators are used to perform arithmetic operations between two operands.

Operator	Description
+ (addition)	Add two operands
- (subtraction)	Subtract right operand from the left
*(multiplication)	Multiply two operands
/ (divide)	Divide left operand by the right one (always results into float)
%(remainder)	Modulus - remainder of the division of left operand by the right

// (floor division) results	Floor division - division that into whole number adjusted to the left in the number line
** (exponent)	Exponent - left operand raised to the power of right

Example:

```
x =20
y =10
print('x + y =',x+y)
print('x - y =',x-y)
print('x * y =',x*y)
print('x / y =',x/y)
print('x // y =',x//y)
print('x ** y =',x**y)
```

Output:

```
x + y =30
x - y =10
x * y =200
x / y =2
x // y =0
x ** y =8
```

2 Assignment Operators

The assignment operators are used to assign the value of the right expression to the left operand.

Operator	Description
= (Assigns to)	Assigns values from right side operands to left side operand.
+= (Assignment after Addition)	It adds right operand to the left operand and assign the result to left operand.
-= (Assignment after Subtraction)	It subtracts right operand from the left operand and assign the result to left operand.
*= (Assignment after Multiplication)	It multiplies right operand with the left operand and assign the result to left operand.
/= (Assignment after Division)	It divides left operand with the right operand and assign the result to left operand.
%= (Assignment after Modulus)	It takes modulus using two operands and assign the result to left operand.
**= (Assignment after Exponent)	Performs exponential (power) calculation on operators and assign value to the left operand.
//= (Assignment after floor division)	It performs floor division on operators and assign value to the left operand.

Example:

Output:

3 Comparison operators

Comparison operators are used to comparing the value of the two operands and returns boolean true or false accordingly.

Operator	Description
== (Equal to)	Equal to - True if both operands are equal.
!= (Not equal to)	Not equal to - True if operands are not equal.
<= (Less than or equal)	Less than or equal to - True if left operand is less than or equal to the right
>= (Greater than or equal)	Greater than or equal to - True if left operand is greater than or equal to the right
< (Less than)	Less than - True if left operand is less than the right
> (Greater than)	Greater than - True if left operand is greater than the right

Example:

```
x =20
y =10
print('x > y is',x > y)
print('x < y is',x < y)
print('x == y is',x == y)
print('x != y is',x != y)
print('x >= y is',x >= y)
print('x <= y is',x <= y)
```

Output:

```
x >y isTrue
x <y isFalse
x == y isFalse
x != y isTrue
x >= y isTrue
x >= y isFalse
```

4 Bitwise Operators

Operator	Description
& (binary and)	Sets each bit to 1 if both bits are 1
(binary or)	Sets each bit to 1 if one of two bits is 1
^ (binary xor)	Sets each bit to 1 if only one of two bits is 1
~ (negation)	Inverts all the bits
<< (left shift)	Shift left by pushing zeros in from the right and let the leftmost bits fall off
>> (right shift)	Shift right by pushing copies of the leftmost bit in from the left, and let the rightmost bits fall off

The bitwise operators perform bit by bit operation on the values of the two operands.

Example:

```
a =9# equal to 1001
b =12# equal to 1100
```

```
print("a & b = ",a& b)
```

```
print("a | b = ",a| b)
```

```
print("a ^ b = ",a ^ b)
```

```
print("~a = ",~a)
```

```
print("a << 2 = ",a <<2)
```

```
print("a >> 2 = ",a >>2)
```

Output:

```
a & b =8
```

```
a | b =13
```

```
a ^ b =5
```

```
~a =-10
```

```
a <<2=36
```

```
a >>2=2
```

5 Identity Operators

Identity operators are used to compare the objects, not if they are equal, but if they are actually the same object, with the same memory location.

Operator	Description
is	Returns true if both variables are the same object (if id(x) equals id(y))
is not	Returns true if both variables are not the same object

Example:

```
x=10
```

```
y=10
```

```
print(x is y)
```

```
print("id(x)= %d id(y)= %d"%(id(x),id(y)))
```

```
y=20
```

```
print(x isnot y)
```

```
print("id(x)= %d id(y)= %d"%(id(x),id(y)))
```

Output:

```
True
```

```
id(x)=10105376id(y)=10105376
```

```
True
```

```
id(x)=10105376id(y)=10105696
```

6 Logical Operators

The logical operators are used primarily in the expression evaluation to make a decision.

Operator	Description
and (logical and)	Returns True if both statements are true
or (logical or)	Returns True if one of the statements is true
not (logical not)	Reverse the result, returns False if the result is true

Example:

```
x=True  
y=False  
print('x and y is',x and y)  
print('x or y is',x or y)  
print('not x is',not x)
```

Output:

```
x and y isFalse  
x or y isTrue  
not x isFalse
```

7 Membership Operators

The logical operators are used primarily in the expression evaluation to make a decision.

Operator	Description
in	it Returns True if a sequence with the specified value is present in the object (list, tuple, or dictionary)
not in)	it Returns True if a sequence with the specified value is not present in the object (list, tuple, or dictionary)

Example:

```
x ='Hello world'  
print('H'in x)  
print('hello'notin x)
```

Output:

```
True  
True
```

Control Flow with Decisions and Loops

Objectives: At the end of this lesson you shall be able to

- if, if-else, if-elif-else.

Conditional Statements in Python performs different computations or actions depending on conditions.

In python, the following are conditional statements

- if
- if - else
- if - elif - else

Indentation

For the ease of programming and to achieve simplicity, python doesn't allow the use of parentheses for the block level code. In Python, indentation is used to declare a block. If two statements are at the same indentation level, then they are the part of the same block.

Generally, four spaces are given to indent the statements which are a typical amount of indentation in python.

Indentation is the most used part of the python language since it declares the block of code. All the statements of one block are intended at the same level indentation.

If statement

The if statement is used to test a specific condition. If the condition is true, a block of code (if-block) will be executed.

Syntax:

```
if expression:  
    statement  
  
if b > a:  
    print("b is greater than a")
```

Output:

b is greater than a

If - else statement

The if-else statement provides an else block combined with the if statement which is executed in the false case of the condition.

If the condition is true, then the if-block is executed. Otherwise, the else-block is executed.

Syntax:

```
if expression:  
    #block of statements  
else:  
    #block of statements
```

#another block of statements

Example:

```
age =int(input("Enter your age : "))  
if age >=18:  
    print("You are eligible to vote !!")  
else:  
    print("Sorry! you have to wait !!")
```

Output:

```
Enter your age:19  
You are eligible to vote!!
```

If - elif - else statement

The elif statement enables us to check multiple conditions and execute the specific block of statements depending upon the true condition among them.

We can have any number of elif statements in our program depending upon our need. However, using elif is optional.

Syntax:

```
if expression 1:  
    # block of statements  
elif expression 2:  
    # block of statements  
elif expression 3:  
    # block of statements  
else:  
    # block of statements
```

Example:

```
marks =int(input("Enter the marks :"))  
if marks >85and marks <=100:  
    print("Congrats! you scored grade A..")  
elif marks >60and marks <=85:  
    print("You scored grade B + ..")  
elif marks >40and marks <=60:  
    print("You scored grade B ..")  
elif(marks >30and marks <=40):  
    print("You scored grade C ..")  
else:  
    print("Sorry you are fail ?")
```

Output:

Enter the marks:70
You scored grade B +..

Example Program:

Aim: python program to find maximum between three numbers

Example:

```
a=int(input("Enter a value : "))
b=int(input("Enter b value : "))
c=int(input("Enter c value : "))
if(a>b)and(a>c):
```

```
print("Maximum value is :",a)
```

```
elif(b>c):
```

```
print("Maximum value is :",b)
```

```
else:
```

```
print("Maximum value is :",c)
```

Output:

```
Enter a value:10
Enter b value:14
Enter c value:9
Maximum value is:14
```

Loop Statements in Python

Objectives: At the end of this lesson you shall be able to

- **while loop, for loop.**

Sometimes we may need to alter the flow of the program. If the execution of a specific code may need to be repeated several numbers of times then we can go for loop statements.

For this purpose, the python provide various types of loops which are capable of repeating some specific code several numbers of times. Those are,

- while loop
- for loop

While loop

With the while loop we can execute a set of statements as long as a condition is true. The while loop is mostly used in the case where the number of iterations is not known in advance.

Syntax:

```
while expression:
    Statement(s)
```

Example:

```
i=1;
while i<=3:
    print(i);
    i=i+1;
```

Output:

```
1
2
3
```

Syntax:

```
while expression:
    Statements
else:
    Statements
```

Example:

```
i=1;
while i<=3:
    print(i)
    i=i+1;
else:print("The while loop terminated");
```

Output:

```
1
2
3
```

The while loop terminated

For loop

The for loop in Python is used to iterate the statements or a part of the program several times. It is frequently used to traverse the data structures like list, tuple, or dictionary.

Syntax:

```
for iterating_var in sequence:
    statement(s)
```

Example:

```
i=1
n=int(input("Enter n value : "))
for i in range(i,n+1):
    print(i,end = ' )
```

Using else with while loop

Python enables us to use the while loop with the else block also. The else block is executed when the condition given in the while statement becomes false.

Output:

```
Enter n value:5
```

```
1 2 3 4 5
```

Using else with for loop

Python allows us to use the else statement with the for loop which can be executed only when all the iterations are exhausted. Here, we must notice that if the loop contains any of the break statement then the else statement will not be executed.

Syntax:

```
for iterating_var in sequence:
```

```
statements
```

```
else:
```

```
statements
```

Example:

```
for i in range(0,5):
    print(i)
else:
    print("for loop completely exhausted");
```

Output:

```
0
1
2
3
4
for loop completely exhausted
```

Note

The range() function returns a sequence of numbers, starting from 0 by default, and increments by 1 (by default), and ends at a specified number.

Example:

range(6) means the values from 0 to 5.

range(2,6) means the values from 2 to 5.

Example Program:

Aim: Program to printing the table of the given number

Example:

```
i=1;
num =int(input("Enter a number:"));
for i in range(1,11):
    print("%d X %d = %d"%(num,i,num*i))
```

Output:

```
Enter a number:10
```

```
10 X 1=10
```

```
10 X 2=20
```

```
10 X 3=30
```

```
10 X 4=40
```

```
10 X 5=50
```

```
10 X 6=60
```

```
10 X 7=70
```

```
10 X 8=80
```

```
10 X 9=90
```

```
10 X 10=100
```

Nested for loop in python

Python allows us to nest any number of for loops inside a for loop. The inner loop is executed n number of times for every iteration of the outer loop.

Syntax:

```
for iterating_var1 in sequence:
```

```
for iterating_var2 in sequence:
```

```
#block of statements
```

Example Program:

Aim: Program to printing the rows of stars of the given number

Example:

```
n =int(input("Enter the no.of rows you want to print : "))
for i in range(0,n):
    for j in range(0,i+1):
        print("*",end="")
    print()
```

Output:

Enter the no.of rows you want to print:4

```

*
*
*
*
*
```

Document and Structure Code

Objectives: At the end of this lesson you shall be able to

-

List Sequence in Python

In python, a list can be defined as a collection of values or items of different types. The items in the list are separated with the comma (,) and enclosed with the square brackets [].

Syntax:

```
List =[value1, value2, value3,...]
```

Example:

```
L0 =[ ]           #creates empty list
L1 =[123,"python",3.7]
L2 =[1,2,3,4,5,6]
L3 =[ "C Programing", "Java", "Python"]
print(L0)
print(L1)
print(L2)
print(L3)
```

List Operators

Operator	Description
+	It is known as concatenation operator used to concatenate two lists
*	It is known as repetition operator. It concatenates the multiple copies of the same list.
[]	It is known as slice operator. It is used to access the list item from list.
[:]	It is known as range slice operator. It is used to access the range of list items from list.
in	It is known as membership operator. It returns if a particular item is present in the specified list.
not in	It is also a membership operator and It returns true if a particular list item is not present in the list.

Example:

```
num=[1,2,3,4,5]
lang=['python','c','java','php']
print(num+lang)
print(num*2)
print(lang[2])
print(lang[1:4])
print('cpp'in lang)
print(6notin num)
```

Output:

```
[1,2,3,4,5,'python','c','java','php']
[1,2,3,4,5,1,2,3,4,5]
```

Output:

```
[ ]
[123,'python',3.7]
[1,2,3,4,5,6].
['C Programing','Java','Python']
```

List indexing

The indexing are processed in the same way as it happens with the strings. The elements of the list can be accessed by using the slice operator [].

The index starts from 0, the first element of the list is stored at the 0th index, the second element of the list is stored at the 1st index, and so on.

Python provides us the flexibility to use the negative indexing also. The negative indices are counted from the right. The last element (right most) of the list has the index -1, its adjacent left element is present at the index -2 and so on until the left most element is encountered.

```
java
['c','java','php']
False
True
```

How to add or change elements to a list?

Python allows us to modify the list items by using the slice and assignment operator.

We can use assignment operator (=) to change an item or a range of items.

Example:

```
num=[1,2,3,4,5]
print(num)
num[2]=30
```

```
print(num)
num[1:3]=[25,36]
print(num)
num[4]="Python"
print(num)
```

Output:

```
[1,2,3,4,5]
[1,2,30,4,5]
[1,25,36,4,5]
[1,25,36,4,'Python']
```

How to delete or remove elements from a list?

Python allows us to delete one or more items in a list by using the del keyword.

Example:

```
num=[1,2,3,4,5]
print(num)
delnum[1]
print(num)
delnum[1:3]
print(num)
```

Output:

```
[1,2,3,4,5]
[1,3,4,5]
[1,5]
```

List Functions

Python provides the following built-in functions which can be used with the lists.

- len()
- max()
- min()
- list()
- sum()
- sorted()
- append()
- remove()
- sort()
- reverse()
- count()
- index()
- insert()
- pop()
- clear()

len()

In Python len() is used to find the length of list,i.e it returns the number of items in the list.

Syntax:

```
len(list)
```

Example:

```
num=[1,2,3,4,5,6]
print("length of list:",len(num))
```

Output:

```
length of list:6
```

?max()

In Python max() is used to find maximum value in the list

Syntax:

```
max(list)
```

Example:

```
num=[1,2,3,4,5,6]
lang=['java','c','python','cpp']
print("Max of list :",max(num))
print("Max of list :",max(lang))
```

Output:

```
Max of list:6
```

```
Max of list: python
```

min()

In Python min() is used to find minimum value in the list

Syntax:

```
min(list)
```

Example:

```
num=[1,2,3,4,5,6]
lang=['java','c','python','cpp']
print("Min of list :",min(num))
print("Min of list :",min(lang))
```

Output:

```
Min of list:1
```

```
Min of list: c
```

sum()

In python, sum(list) function returns sum of all values in the list. List values must in number type.

Syntax:

```
sum(list)
```

Example:

```
num=[1,2,3,4,5,6]
print("sum of list items :",sum(num))
```

Output:

```
sum of list items:21
```

sorted()

In python, sorted(list) function is used to sort all items of list in an ascending order.

Syntax:

```
sorted(list)
```

Example:

```
num=[1,3,2,4,6,5]
lang=['java','c','python','cpp']
print(sorted(num))
print(sorted(lang))
```

Output:

```
[1,2,3,4,5,6]
['c','cpp','java','python']
```

list()

The list() method takes sequence types and converts them to lists. This is used to convert a given string or tuple into list.

Syntax:

```
list(sequence)
```

Example:

```
str="python"
list1=list(str)
print(list1)
```

Output:

```
['p','y','t','h','o','n']
```

append ()

In python append() method adds an item to the end of the list.

Syntax:

```
list.append(item)
item may be number,string,list and etc
```

Example:

```
num=[1,2,3,4,5]
lang=['python','c','java','php']
num.append(6)
print(num)
lang.append("cpp")
print(lang)
```

Output:

```
[1,2,3,4,5,6]
['python','c','java','php','cpp']
```

remove()

In python remove() method removes the first item from the list which is equal to the passed value. It throws an error if the item is not present in the list.

Syntax:

```
list.remove(item)
```

item may be number,string,list and etc

Example:

```
num=[1,2,3,4,5]
lang=['python','c','java','php','c']
num.remove(2)
print(num)
lang.remove("c")# first occurrence will remove
print(lang)
lang.remove("cpp")
print(lang)
```

Output:

```
[1,3,4,5]
['python','java','php','c']
ValueError:list.remove(x):x not in list
```

sort()

In python sort() method sorts the list elements. It also sorts the items into descending and ascending order. It takes an optional parameter 'reverse' which sorts the list into descending order. By default, list sorts the elements into ascending order.

Syntax:

```
list.sort([reverse=true])
reverse, which displays items in descending
```

Example:

```
lang =['p','y','t','h','o','n']# Char list
even =[6,8,2,4,10]# int list
print(lang)
print(even)
lang.sort()
even.sort()
print("\nAfter Sorting:\n")
print(lang)
print(even)
print("In Descending Order :\n")
even.sort(reverse=True)
print(even)
```

Output:

```
[‘p’, ‘y’, ‘t’, ‘h’, ‘o’, ‘n’]
```

```
[6,8,2,4,10]
```

After Sorting:

```
[‘h’, ‘n’, ‘o’, ‘p’, ‘t’, ‘y’]
```

```
[2,4,6,8,10]
```

In Descending Order :

```
[10,8,6,4,2]
```

reverse()

In python reverse() method reverses elements of the list. If the list is empty, it simply returns an empty list. After reversing the last index value of the list will be present at 0 index.

Syntax:

```
list.reverse()
```

Example:

```
lang = [‘p’, ‘y’, ‘t’, ‘h’, ‘o’, ‘n’]
```

```
lang = [‘p’, ‘y’, ‘t’, ‘h’, ‘o’, ‘n’]
```

```
print(“After reverse”)
```

```
lang.reverse()
```

```
print(lang)
```

Output:

After reverse

```
[‘n’, ‘o’, ‘h’, ‘t’, ‘y’, ‘p’]
```

count()

In python count() method returns the number of times element appears in the list. If the element is not present in the list, it returns 0.

Syntax:

```
list.count(item)
```

Example:

```
num=[1,2,3,4,3,2,2,1,3,4,5,7,8]
```

```
cnt=num.count(2)
```

```
print(“Count of 2 is:”,cnt)
```

```
cnt=num.count(10)
```

```
print(“Count of 10 is:”,cnt)
```

Output:

Count of 2is:3

Count of 10is:0

index()

In python index () method returns index of the passed element. This method takes an argument and returns index of it. If the element is not present, it raises a ValueError.

If list contains duplicate elements, it returns index of first occurred element.

This method takes two more optional parameters start and end which are used to search index within a limit.

Syntax:

```
list.index(x[, start[, end]])
```

Example:

```
lang =[‘p’, ‘y’, ‘t’, ‘h’, ‘o’, ‘n’, ‘p’, ‘r’, ‘o’, ‘g’, ‘r’, ‘a’, ‘m’]
```

```
print(“index of t is:”,lang.index(‘t’))
```

```
print(“index of p is:”,lang.index(‘p’))
```

```
print(“index of p is:”,lang.index(‘p’,3,10))
```

```
print(“index of p is:”,lang.index(‘z’))
```

Output:

index of t is:2

index of p is:0

index of p is:6

ValueError:‘z’isnotinlist

insert()

In python insert() method inserts the element at the specified index in the list. The first argument is the index of the element before which to insert the element.

Syntax:

```
list.insert(i,x)
```

i : index at which element would be inserted.

x : element to be inserted.

Example:

```
num=[10,20,30,40,50]
```

```
num.insert(4,60)
```

```
print(“updated list is :”,num)
```

```
num.insert(7,70)
```

```
print(“updated list is :”,num)
```

Output:

updated listis:[10,20,30,40,60,50]

updated listis:[10,20,30,40,60,50,70]

pop()

In python pop() element removes an element present at specified index from the list. It returns the popped element.

Syntax:

```
list.pop([i])
```

Example:

```
num=[10,20,30,40,50]
```

```
num.pop()
```

```
print(“updated list is :”,num)
```

```

num.pop(2)
print("updated list is :",num)
num.pop(7)
print("updated list is :",num)

```

Output:

```

updated list is:[10,20,30,40]
updated list is:[10,20,40]
IndexError: pop index out of range

```

clear()

In python clear() method removes all the elements from the list. It clears the list completely and returns nothing.

Tuple Sequence in Python

Objectives: At the end of this lesson you shall be able to

- tuple
- indexing:
- Tuple Operators
- How to add or remove elements to a list?
- Tuple Functions.

In Python Tuple is used to store the sequence of immutable python objects. Tuple is similar to lists since the value of the items stored in the list can be changed whereas the tuple is immutable and the value of the items stored in the tuple cannot be changed.

A tuple can be written as the collection of comma-separated values enclosed with the small brackets.

Syntax:

```
Tuple=(value1, value2...)
```

Example:

```

T1=()
T2=(10,30,20,40,60)
T3=("C","Java","Python")
T4=(501,"abc",19.5)
T5=(90,)
print(T1)
print(T2)
print(T3)
print(T4)
print(T5)

```

Syntax:

```

list.clear()
Example:
num=[10,20,30,40,50]
num.clear()
print("After clearing ",num)

```

Output:

```
After clearing [ ]
```

Output:

```

()
(10,30,20,40,60)
('C','Java','Python')
(501,'abc',19.5)
(90,)

```

Tuple indexing:

The indexing and slicing in tuple are similar to lists. The indexing in the tuple starts from 0 and goes to length (tuple) - 1.

The items in the tuple can be accessed by using the slice operator. Python also allows us to use the colon operator to access multiple items in the tuple.

Unlike other languages, python provides us the flexibility to use the negative indexing also. The negative indices are counted from the right. The last element (right most) of the tuple has the index -1, its adjacent left element is present at the index -2 and so on until the left most elements is encountered.

Tuple Operators

Operator	Description
+	It is known as concatenation operator used to concatenate two tuples
*	It is known as repetition operator. It concatenates the multiple copies of the same tuple.
[]	It is known as slice operator. It is used to access the item from tuple.
[:]	It is known as range slice operator. It is used to access the range of items from tuple.
in	It is known as membership operator. It returns if a particular item is present in the specified tuple.
not in	It is also a membership operator and It returns true if a particular item is not present in the tuple.

Example:

```

num=(1,2,3,4,5)
lang=('python','c','java','php')
print(num+lang)
print(num*2)
print(lang[2])
print(lang[1:4])
print('cpp'in lang)
print(6notin num)

```

Output:

```

(1,2,3,4,5,'python','c','java','php')
(1,2,3,4,5,1,2,3,4,5)
java
('c','java','php')
True
True

```

How to add or remove elements to a list?

Unlike lists, the tuple items cannot be updated or deleted as tuples are immutable. To delete an entire tuple, we can use the del keyword with the tuple name.

Example:

```

mytuple=('python','c','java','php')
mytuple[3]="html"
#tuple' object does not support item assignment
print(mytuple)
delmytuple[3]
# 'tuple' object doesn't support item deletion
print(mytuple)
del mytuple
#deletes entire tuple

```

Output:

```

'tuple'object does not support item assignment
'tuple'object doesn't support item deletion

```

Tuple Functions

Python provides the following built-in functions which can be used with the tuples.

- len()
- max()
- min()
- tuple()
- sum()
- sorted()
- index()
- count()

len()

In Python len() is used to find the length of tuple,i.e it returns the number of items in the tuple.

Syntax:

```
len(tuple)
```

Example:

```

num=(1,2,3,4,5,6)
print("length of tuple:",len(num))

```

Output:

```
length of tuple:6
```

max()

In Python max() is used to find maximum value in the tuple.

Syntax:

```
max(tuple)
```

Example:

```

num=(1,2,3,4,5,6)
lang= ('java','c','python','cpp')
print("Max of tuple :",max(num))
print("Max of tuple :",max(lang))

```

Output:

```

Max of tuple:6
Max of tuple: python

```

min()

In Python min() is used to find minimum value in the tuple.

Syntax:

```
min(tuple)
```

Example:

```

num=(1,2,3,4,5,6)
lang= ('java','c','python','cpp')
print("Min of tuple :",min(num))
print("Min of tuple :",min(lang))

```

Output:

```

Min of tuple:1
Min of tuple: c

```

sum()

In python, sum(tuple) function returns sum of all values in the tuple. Tuple values must in number type.

Syntax:

```
sum(tuple)
```

Example:

```

num=(1,2,3,4,5,6)
print("sum of tuple items :",sum(num))

```

Output:

sum of tuple items:21

sorted()

In python, sorted (tuple) function is used to sort all items of tuple in an ascending order. It also sorts the items into descending and ascending order. It takes an optional parameter 'reverse' which sorts the tuple into descending order.

Syntax:

```
sorted(tuple[, reverse=True])
```

Example:

```
num=(1,3,2,4,6,5)
lang=('java','c','python','cpp')
print(sorted(num))
print(sorted(lang))
print(sorted(num,reverse=True))
```

Output:

```
(1,2,3,4,5,6)
('c','cpp','java','python')
(6,5,4,3,2,1)
```

tuple (sequence)

The tuple() method takes sequence types and converts them to tuples. This is used to convert a given string or list into tuple.

Syntax:

```
tuple(sequence)
```

Example:

```
str="python"
tuple1=tuple(str)
print(tuple1)
num=[1,2,3,4,5,6]
tuple2=tuple(num)
print(tuple2)
```

Output:

```
('p','y','t','h','o','n')
(1,2,3,4,5,6)
```

Set Sequence in Python

Objectives: At the end of this lesson you shall be able to

- **creating a set**
- **set Operators**
- **set Functions.**

In python, the set can be defined as the unordered collection of various items enclosed within the curly braces. The elements of the set cannot be duplicate. The elements of the python set must be immutable.

count()

In python count() method returns the number of times element appears in the tuple. If the element is not present in the tuple, it returns 0.

Syntax:

```
tuple.count(item)
```

Example:

```
num=(1,2,3,4,3,2,2,1,3,4,5,7,8)
cnt=num.count(2)
print("Count of 2 is:",cnt)
cnt=num.count(10)
print("Count of 10 is:",cnt)
```

Output:

```
Count of 2 is:3
Count of 10 is:0
```

Index()

In python index () method returns index of the passed element. This method takes an argument and returns index of it. If the element is not present, it raises a ValueError.

If tuple contains duplicate elements, it returns index of first occurred element.

This method takes two more optional parameters start and end which are used to search index within a limit.

Syntax:

```
tuple.index(x[, start[, end]])
```

Example:

```
lang=('p','y','t','h','o','n','p','r','o','g','r','a','m')
print("index of t is:",lang.index('t'))
print("index of p is:",lang.index('p'))
print("index of p is:",lang.index('p',3,10))
print("index of p is:",lang.index('z'))
```

Output:

```
index of t is:2
index of p is:0
index of p is:6
ValueError:'z'isnotintuple.
```

Unlike other collections in python, there is no index attached to the elements of the set, i.e., we cannot directly access any element of the set by the index. However, we can print them all together or we can get the list of elements by looping through the set.

Creating a set

The set can be created by enclosing the comma separated items with the curly braces.

Syntax:

```
Set={value1, value2,...}
```

Example:

```
Days ={"Monday", "Tuesday", "Wednesday ", "Thursday",  
"Friday", "Saturday", "Sunday"}
```

```
print(Days)  
print(type(Days))  
print("Looping through the set elements ... ")  
for i in Days:  
    print(i)
```

Output:

```
{'Wednesday', 'Tuesday', 'Sunday', 'Friday', 'Thursday',  
'Saturday', 'Monday'}
```

```
Looping through the set elements...  
Wednesday  
Tuesday  
Sunday  
Friday  
Thursday  
Saturday  
Monday
```

Set Operators

In Python, we can perform various mathematical operations on python sets like union, intersection, difference, etc

Operator	Description
	Union Operator
&	Intersection Operator
-	Difference Operator:

Union (|) Operator

The union of two sets are calculated by using the or (|) operator. The union of the two sets contains the all the items that are present in both the sets.

Example:

```
Days1={"Mon", "Tue", "Wed", "Sat"}  
Days2={"Thr", "Fri", "Sat", "Sun", "Mon"}  
print(Days1 | Days2)
```

Output:

```
{'Thr', 'Fri', 'Sun', 'Tue', 'Wed', 'Mon', 'Sat'}
```

Intersection (&) Operator

The & (intersection) operator is used to calculate the intersection of the two sets in python. The intersection of the two sets are given as the set of the elements that common in both sets.

Example:

```
Days1={"Mon", "Tue", "Wed", "Sat"}  
Days2={"Thr", "Fri", "Sat", "Sun", "Mon"}  
print(Days1 & Days2)
```

Output:

```
{'Mon', 'Sat'}
```

Difference (-) Operator

The difference of two sets can be calculated by using the subtraction (-) operator. The resulting set will be obtained by removing all the elements from set 1 that are present in set 2.

Example:

```
Days1={"Mon", "Tue", "Wed", "Sat"}  
Days2={"Thr", "Fri", "Sat", "Sun", "Mon"}  
print(Days1 - Days2)
```

Output:

```
{'Tue', 'Wed'}
```

Set Functions

Python contains the following methods to be used with the sets. Those are

- len(set)
- max(set)
- min(set)
- sum(set)
- sorted(set)
- set()
- add()
- update()
- discard()
- remove()
- pop()
- clear()
- union()
- intersection()
- difference()
- issubset()
- issuperset()

len()

In Python len() is used to find the length of set,i.e it returns the number of items in the set.

Syntax:

```
len(set)
```

Example:

```
num={1,2,3,4,5,6}
print("length of set:",len(num))
```

Output:

length of set:6

max()

In Python max() is used to find maximum value in the set.

Syntax:

```
max(set)
```

Example:

```
num={1,2,3,4,5,6}
lang={'java','c','python','cpp'}
print("Max of set :",max(num))
print("Max of set :",max(lang))
```

Output:

Max of set:6

Max of set: python

min()

In Python min() is used to find minimum value in the set.

Syntax:

```
min(set)
```

Example:

```
num={1,2,3,4,5,6}
lang={'java','c','python','cpp'}
print("Min of set :",min(num))
print("Min of set :",min(lang))
```

Output:

Min of set:1

Min of set: c

sum()

In python, sum(set) function returns sum of all values in the set. Set values must in number type.

Syntax:

```
sum(set)
```

Example:

```
num={1,2,3,4,5,6}
print("sum of set items :",sum(num))
```

Output:

sum of set items:21

sorted()

In python, sorted (set) function is used to sort all items of set in an ascending order. It also sorts the items into descending and ascending order. It takes an optional parameter 'reverse' which sorts the set into descending order.

Syntax:

```
sorted(set[,reverse=True])
```

Example:

```
num={1,3,2,4,6,5}
lang={'java','c','python','cpp'}
print(sorted(num))
print(sorted(lang))
print(sorted(num,reverse=True))
```

Output:

```
{1,2,3,4,5,6}
{'c','cpp','java','python'}
{6,5,4,3,2,1}
set()
```

The set() method takes sequence types and converts them to sets. This is used to convert a given string or list or tuple into set.

Syntax:

```
set(sequence)
```

Example:

```
set1=set("PYTHON")
print(set1)
days=["Mon","Tue","Wed","Thur","Fri","Sat","Sun"]
set2 =set(days)
print(set2)
days=("Mon","Tue","Wed","Thur","Fri","Sat","Sun")
set3 =set(days)
print(set3)
```

Output:

```
{'N','O','T','H','P','Y'}
{'Fri','Thur','Tue','Sun','Mon','Sat','Wed'}
{'Fri','Thur','Tue','Sun','Mon','Sat','Wed'}
```

add()

In python, the add() method used to add some particular item to the set.

Syntax:

```
set.add (item)
```

Example:

```

Days = {"Monday", "Tuesday", "Wednesday",
        "Thursday", "Friday"}
print("\n printing the original set ... ")
print(Days)
Days.add("Saturday");
Days.add("Sunday");
print("\n Printing the modified set... ");
print(Days)

```

Output:

```

printing the original set...
{'Wednesday', 'Friday', 'Thursday', 'Tuesday', 'Monday'}
Printing the modified set...
{'Wednesday', 'Sunday', 'Friday', 'Thursday', 'Tuesday',
 'Saturday', 'Monday'}

```

update()

Python provides the update () method to add more than one item in the set.

Syntax:

```
set.update([item1, item2...])
```

Example:

```

Months={"Jan", "Feb", "Mar", "Apr"}
print("\n Printing the original set ... ")
print(Months)
Months.update(["May", "Jun", "Jul"])
print("\n Printing the modified set... ");
print(Months)

```

Output:

```

Printing the original set...
{'Mar', 'Apr', 'Jan', 'Feb'}
Printing the modified set...
{'Mar', 'Apr', 'Jan', 'Jun', 'May', 'Jul', 'Feb'}

```

discard()

Python provides discard () method which can be used to remove the items from the set. If item doesn't exist in the set, the python will not give the error. The program maintains its control flow.

Syntax:

```
set.discard(item)
```

Example:

```

Months={"Jan", "Feb", "Mar", "Apr"}
print("\n printing the original set ... ")
print(Months)

```

```
Months.discard("Apr")
```

```

print("\n Printing the modified set... ");
print(Months)
Months.discard("May")      #doesn't give error
print("\n Printing the modified set... ");
print(Months)

```

Output:

```

printing the original set...
{'Jan', 'Apr', 'Mar', 'Feb'}
Printing the modified set...
{'Jan', 'Mar', 'Feb'}
Printing the modified set...
{'Jan', 'Mar', 'Feb'}

```

remove()

Python provides remove () method which can be used to remove the items from the set. If item doesn't exist in the set, the python will give the error.

Syntax:

```
set.remove(item)
```

Example:

```

Months={"Jan", "Feb", "Mar", "Apr"}
print("\n printing the original set ... ")
print(Months)
Months.remove("Apr")
print("\n Printing the modified set... ");
print(Months)
Months.remove("May")      #it give error
print("\n Printing the modified set... ");
print(Months)

```

Output:

```

printing the original set...
{'Feb', 'Jan', 'Apr', 'Mar'}
Printing the modified set...
{'Feb', 'Jan', 'Mar'}
KeyError:'May' doesn't exist.

```

pop()

In Python, pop () method is used to remove the item. However, this method will always remove the last item.

Syntax:

```
set.pop()
```

Example:

```

Days = {"Monday", "Tuesday", "Wednesday",
        "Thursday", "Friday"}
print("\n printing the original set ... ")
print(Days)
Days.pop()
print("\n Printing the modified set... ");
print(Days)

```

Output:

Printing the original set...

{'Monday', 'Wednesday', 'Friday', 'Tuesday', 'Thursday'}

Printing the modified set...

{'Wednesday', 'Friday', 'Tuesday', 'Thursday'}

clear()

In Python, clear () method is used to remove the all items in set.

Syntax:

```
set.clear()
```

Example:

```

Days = {"Monday", "Tuesday", "Wednesday",
        "Thursday", "Friday"}
print("\n printing the original set ... ")
print(Days)
Days.clear()
print("\n Printing the modified set... ");
print(Days)

```

Output:

printing the original set...

{'Monday', 'Wednesday', 'Friday', 'Tuesday', 'Thursday'}

Printing the modified set...

```
set()
```

union ()

In Python, the union () method is used to perform union of two sets. The union of the two sets contains the all the items that are present in both the sets.

Syntax:

```
set1.union (set2)
```

Example:

```

Days1={"Mon", "Tue", "Wed", "Sat"}
Days2={"Thr", "Fri", "Sat", "Sun", "Mon"}
print(Days1.union(Days2))

```

Output:

{'Thr', 'Fri', 'Sun', 'Tue', 'Wed', 'Mon', 'Sat'}

intersection ()

In Python, the intersection () is used to calculate the intersection of the two sets in python. The intersection of the two sets is given as the set of the elements that common in both sets.

Syntax:

```
set1.intersection (set2)
```

Example:

```

Days1={"Mon", "Tue", "Wed", "Sat"}
Days2={"Thr", "Fri", "Sat", "Sun", "Mon"}
print(Days1.intersection(Days2))

```

Output:

{'Mon', 'Sat'}

difference ()

The difference of two sets can be calculated by using the difference () method. The resulting set will be obtained by removing all the elements from set 1 that are present in set 2.

Syntax:

```
set1.difference (set2)
```

Example:

```

Days1={"Mon", "Tue", "Wed", "Sat"}
Days2={"Thr", "Fri", "Sat", "Sun", "Mon"}
print(Days1.difference(Days2))

```

Output:

{'Tue', 'Wed'}

issubset()

The issubset() method returns True if all elements of a set are present in another set (passed as an argument). If not, it returns False.

Syntax:

```
set1.issubset (set2)
```

Example:

```

set1={1,2,3,4}
set2={1,2,3,4,5,6,7,8,9}
print(set1.issubset(set2))
print(set2.issubset(set1))

```

Output:

True

False

issuperset()

The issuperset () method returns True if a set has every elements of another set (passed as an argument). If not, it returns False.

Syntax:

```
set1.issuperset (set2)
```

Example:

```
set1={1,2,3,4}
```

```
set2={1,2,3,4,5,6,7,8,9}
```

```
print(set1.issuperset(set2))
```

```
print(set2.issuperset(set1))
```

Output:

```
False
```

```
True
```

Dictionary Sequence in Python

Objectives: At the end of this lesson you shall be able to

- accessing the dictionary values
- updating dictionary values
- deleting dictionary values
- accessing the dictionary values using loops
- dictionary Functions.

In python a dictionary is the collection of key-value pairs where the value can be any python object whereas the keys are the immutable python object, i.e., Numbers, string or tuple.

The dictionary can be created by using multiple key-value pairs which are separated by comma(,) and enclosed within the curly braces {}.

Syntax:

```
Dict={key1:value1, key2:value2, .....}
```

Example:

```
student ={"Name":"Kiran","Age":22,"Regno": 562,  
"Branch":"CSE"}  
print(student)
```

Output:

```
{'Name':'Kiran','Age':22,'Regno':562,'Branch':'CSE'}
```

Accessing the dictionary values

The data can be accessed in the list and tuple by using the indexing.

However, the values can be accessed in the dictionary by using the keys as keys are unique in the dictionary.

Example:

```
student ={"Name":"Kiran","Age":22,"Regno": 562,  
"Branch":"CSE"}  
print("Name : ",student["Name"])  
print("Age : ",student["Age"])  
print("RegNo : ",student["Regno"])  
print("Branch : ",student["Branch"])
```

Output:

```
Name : Kiran
```

```
Age :22
```

```
RegNo :562
```

```
Branch : CSE
```

Updating dictionary values

The dictionary is a mutable data type, and its values can be updated by using the specific keys.

Example:

```
student ={"Name":"Kiran","Age":22,"Regno": 562,  
"Branch":"CSE"}  
print("printing student data .... ")  
print(student)  
student["Name"]="Kishore"  
print("printing updated data .... ")  
print(student)
```

Output:

```
printing student data ....
```

```
{'Name':'Kiran','Age':22,'Regno':562,'Branch':'CSE'}
```

```
printing updated data ....
```

```
{'Name':'Kishore','Age':22,'Regno':562,'Branch':'CSE'}
```

Deleting dictionary values

The items of the dictionary can be deleted by using the del keyword.

Example:

```
student ={"Name":"Kiran","Age":22,"Regno": 562,  
"Branch":"CSE"}  
print("printing student data .... ")  
print(student)  
del student["Branch"]  
print("printing the modified information ")  
print(student)
```

Output:

```
printing student data ....
```

```
{'Name':'Kiran','Age':22,'Regno':562,'Branch':'CSE'}
```

```
printing the modified information
```

```
{'Name':'Kiran','Age':22,'Regno':562}
```

Accessing the dictionary values using loops

A dictionary can be iterated using the for loop. We can able to access only keys, only values and both keys&values.

print all the keys of a dictionary

Example:

```
student ={"Name":"Kiran","Age":22,"Regno": 562,  
"Branch":"CSE"}  
  
print("Keys are :")  
  
for x in student:  
  
    print(x)
```

Output:

Keys are :

Name
Age
Regno
Branch

print all the values of a dictionary

Example:

```
print("values are :")  
  
for x in student:  
  
    print(student[x])
```

Output:

values are :

Kiran
22
562
CSE

print all the Keys & values of a dictionary

Example:

```
print("Key and values are :")  
  
for x,y in student.items():  
  
    print(x,y)
```

Output:

Key and values are :

Name Kiran
Age 22
Regno 562
Branch CSE

Dictionary Functions

Python supports following in-built functions, Those are

- len()
- copy()
- get()
- keys()
- items()
- values()
- update()
- pop()
- clear()

len()

In python, len() function is used to find length of given dictionary.

Syntax:

len(dictionary)

Example:

```
student ={"Name":"Kiran","Age":22,"Regno": 562,  
"Branch":"CSE"}
```

```
print("Length of Dictionary is:",len(student))
```

Output:

Length of Dictionary is:4

copy()

It returns another copy of given dictionary.

Syntax:

dictionary.copy()

Example:

```
student ={"Name":"Kiran","Age":22,"Regno": 562,  
"Branch":"CSE"}
```

```
student2=student.copy()
```

```
print(student2)
```

Output:

```
{'Name':'Kiran','Age':22,'Regno':562,'Branch':'CSE'}
```

get()

In python, get() is used to get the value of specified key from dictionary.

Syntax:

dictionary.get()

Example:

```
student ={"Name":"Kiran","Age":22,"Regno": 562,  
"Branch":"CSE"}
```

```
print("Name is :",student.get("Name"))
```

```
print("RegNo is :",student.get("Regno"))
```

Output:

```
Name is: Kiran
RegNo is:562
```

keys()

In python keys() method is used to fetch all the keys from the dictionary

Syntax:

```
dictionary.keys()
```

Example:

```
student ={"Name":"Kiran","Age":22,"Regno" :562,
"Branch":"CSE"}
for x in student.keys():
    print(x)
```

Output:

```
Name
Age
Regno
Branch
```

items()

In python items() method returns a new view of the dictionary. This view is collection of key value tuples.

Syntax:

```
dictionary.items()
```

Example:

```
student ={"Name":"Kiran","Age":22,"Regno" :562,
"Branch":"CSE"}
for x in student.items():
    print(x)
```

Output:

```
('Name','Kiran')
('Age',22)
('Regno',562)
('Branch','CSE')
```

values()

In python values() method is used to collect all the values from a dictionary.

Syntax:

```
Dictionary.values()
```

Example:

```
student ={"Name":"Kiran","Age":22,"Regno" :562,
"Branch":"CSE"}
for x in student.values():
    print(x)
```

Output:

Output:

```
Kiran
22
562
CSE
```

update()

In python update() method updates the dictionary with the key and value pairs. It inserts key/value if it is not present. It updates key/value if it is already present in the dictionary.

Syntax:

```
Dictionary.update({key:value,....})
```

Example:

```
student ={"Name":"Kiran","Age":22,"Regno" :562,
"Branch":"CSE"}
student.update({"Regno":590})
student.update({"phno":56895})
print(student)
```

Output:

```
{'Name':'Kiran','Age':22,'Regno':590,'Branch':'CSE','phno':56895}
```

pop()

In python pop() method removes an element from the dictionary. It removes the element which is associated to the specified key.

If specified key is present in the dictionary, it remove and return its value.

If the specified key is not present, it throws an error KeyError.

Syntax:

```
Dictionary.remove(key)
```

Example:

```
student ={"Name":"Kiran","Age":22,"Regno" :562,
"Branch":"CSE"}
student.pop('Age')
print(student)
student.pop('hallno')
print(student)
```

Output:

```
{'Name':'Kiran','Regno':562,'Branch':'CSE'}
KeyError:'hallno'
```

clear()

In python, clear() is used to delete all the items of the dictionary.

Syntax:

```
Dictionary.clear()
```

Example:

```
student ={"Name":"Kiran","Age":22,"Regno" :562,  
"Branch":"CSE"}  
print(student)  
student.clear()
```

```
print(student)
```

Output:

```
{'Name':'Kiran', 'Age':22, 'Regno':562, 'Branch':'CSE'}  
{}
```

Python Arrays

Objectives: At the end of this lesson you shall be able to

- elements of an Array.

Note: Python does not have built-in support for Arrays, but Python Lists can be used instead.

Arrays

Note: This page shows you how to use LISTS as ARRAYS, however, to work with arrays in Python you will have to import a library, like the NumPy library.

Arrays are used to store multiple values in one single variable:

Example

Create an array containing car names:
cars = ["Ford", "Volvo", "BMW"]
Try it Yourself "

What is an Array?

An array is a special variable, which can hold more than one value at a time.

If you have a list of items (a list of car names, for example), storing the cars in single variables could look like this:

```
car1 = "Ford"  
car2 = "Volvo"  
car3 = "BMW"
```

However, what if you want to loop through the cars and find a specific one? And what if you had not 3 cars, but 300?

The solution is an array!

An array can hold many values under a single name, and you can access the values by referring to an index number.

Access the Elements of an Array

You refer to an array element by referring to the index number.

Example

Get the value of the first array item:
x = cars[0]
Try it Yourself "

Example

Modify the value of the first array item:
cars[0] = "Toyota"

Try it Yourself

The Length of an Array

Use the len() method to return the length of an array (the number of elements in an array).

Example

Return the number of elements in the cars array:
x = len(cars)
Try it Yourself "

Note: The length of an array is always one more than the highest array index.

Looping Array Elements

You can use the for in loop to loop through all the elements of an array.

Example

Print each item in the cars array:
for x in cars:
 print(x)
Try it Yourself "

Adding Array Elements

You can use the append() method to add an element to an array.

Example

Add one more element to the cars array:
cars.append("Honda")
Try it Yourself "

Removing Array Elements

You can use the pop() method to remove an element from the array.

Example

Delete the second element of the cars array:
cars.pop(1)
Try it Yourself "

Try it Yourself

You can also use the remove() method to remove an element from the array.

Example

Delete the element that has the value "Volvo":

```
cars.remove("Volvo")
```

Try it Yourself "

Note: The list's remove() method only removes the first occurrence of the specified value.

Array Methods

Python has a set of built-in methods that you can use on lists/arrays.

Method	Description
append()	Adds an element at the end of the list
clear()	Removes all the elements from the list
copy()	Returns a copy of the list
count()	Returns the number of elements with the specified value
extend()	Add the elements of a list (or any iterable), to the end of the current list
index()	Returns the index of the first element with the specified value
insert()	Adds an element at the specified position
pop()	Removes the element at the specified position
remove()	Removes the first item with the specified value
reverse()	Reverses the order of the list
sort()	Sorts the list

Note: Python does not have built-in support for Arrays, but Python Lists can be used instead.

Perform Operations Using Modules and Tools

Objectives: At the end of this lesson you shall be able to

- creating module
- loading module
- rename module.

Modules in Python

In python module can be defined as a python program file which contains a python code including python functions, class, or variables. In other words, we can say that our python code file saved with the extension (.py) is treated as the module.

Modules in Python provides us the flexibility to organize the code in a logical way. To use the functionality of one module into another, we must have to import the specific module.

Creating Module

Example: demo.py

```
# Python Module example
def sum(a,b):
    return a+b
defsub(a,b):
    return a-b
defmul(a,b):
    return a*b
defdiv(a,b):
    return a/b
```

In the above example we have defined 4 functions sum(), sub(), mul() and div() inside a module named demo.

Loading the module in our python code

We need to load the module in our python code to use its functionality. Python provides two types of statements as defined below.

- 1 import statement
- 2 from-import statement

1 import statement

The import statement is used to import all the functionality of one module into another. Here, we must notice that we can use the functionality of any python source file by importing that file as the module into another python source file.

We can import multiple modules with a single import statement.

Syntax:

```
import module1,module2..
```

Example:

```
import demo      #importing entire Module
a=int(input("Enter a :"))
b=int(input("Enter b :"))
print("Sum is :",demo.sum(a,b))
print("Sub is :",demo.sub(a,b))
print("Mul is :",demo.mul(a,b))
print("Div is :",demo.div(a,b))
```

Output:

```
Enter a :12
Enter b :6
Sum is:18
Sub is:6
Mul is:72
Div is:2.0
```

2 from-import statement

Instead of importing the whole module into the namespace, python provides the flexibility to import only the specific attributes of a module. This can be done by using from - import statement. In such case we don't use the dot operator.

Syntax:

```
from module-name import*
```

Example:

```
from demo import*
a=int(input("Enter a :"))
b=int(input("Enter b :"))
print("Sum is :",sum(a,b))
print("Sub is :",sub(a,b))
print("Mul is :",mul(a,b))
print("Div is :",div(a,b))
```

Output:

```
Enter a :12
Enter b :6
Sum is:18
Sub is:6
Mul is:72
Div is:2.0
```

We can import specific function from a module without importing the module as a whole. Here is an example.

Syntax:

```
from module-name import function1,function2...
```

Example:

```
from demo import sub,mul  
#importing specific functionality from Module  
a=int(input("Enter a :"))  
b=int(input("Enter b :"))  
print("Sub is :",sub(a,b))  
print("Mul is :",mul(a,b))
```

Output:

```
Enter a :12  
Enter b :6  
Sub is:6  
Mul is:72
```

Python Iterators

Objectives: At the end of this lesson you shall be able to

- **looping Through an Iterator**
- " **create an Iterator**
- " **stopIteration.**

Python Iterators

An iterator is an object that contains a countable number of values.

An iterator is an object that can be iterated upon, meaning that you can traverse through all the values.

Technically, in Python, an iterator is an object which implements the iterator protocol, which consist of the methods `__iter__()` and `__next__()`.

Iterator vs Iterable

Lists, tuples, dictionaries, and sets are all iterable objects. They are iterable containers which you can get an iterator from.

All these objects have a `iter()` method which is used to get an iterator:

Example

Return an iterator from a tuple, and print each value:

```
mytuple = ("apple", "banana", "cherry")  
myit = iter(mytuple)  
print(next(myit))  
print(next(myit))  
print(next(myit))
```

Try it Yourself "

Even strings are iterable objects, and can return an iterator:

Renaming a module

Python provides us the flexibility to import some module with a specific name so that we can use this name to use that module in our python source file.

Syntax:

```
import module-name as specific-name
```

Example:

```
import demo as c  
a=int(input("Enter a :"))  
b=int(input("Enter b :"))  
print("Sum is :",c.sum(a,b))  
print("Sub is :",c.sub(a,b))
```

Output:

```
Enter a :25  
Enter b :12  
Sum is:37  
Sub is:13
```

Example

Strings are also iterable objects, containing a sequence of characters:

```
mystr = "banana"  
myit = iter(mystr)  
print(next(myit))  
print(next(myit))  
print(next(myit))  
print(next(myit))  
print(next(myit))  
print(next(myit))  
print(next(myit))
```

Try it Yourself "

Looping Through an Iterator

We can also use a for loop to iterate through an iterable object:

Example

Iterate the values of a tuple:

```
mytuple = ("apple", "banana", "cherry")  
for x in mytuple:  
    print(x)
```

Try it Yourself "

Example

Iterate the characters of a string:

```
mystr = "banana"  
for x in mystr:  
    print(x)
```

Try it Yourself "

The for loop actually creates an iterator object and executes the next() method for each loop.

Create an Iterator

To create an object/class as an iterator you have to implement the methods `__iter__()` and `__next__()` to your object.

As you have learned in the Python Classes/Objects chapter, all classes have a function called `__init__()`, which allows you to do some initializing when the object is being created.

The `__iter__()` method acts similar, you can do operations (initializing etc.), but must always return the iterator object itself.

The `__next__()` method also allows you to do operations, and must return the next item in the sequence.

Example

Create an iterator that returns numbers, starting with 1, and each sequence will increase by one (returning 1,2,3,4,5 etc.):

```
class MyNumbers:  
    def __iter__(self):  
        self.a = 1  
        return self  
    def __next__(self):  
        x = self.a  
        self.a += 1  
        return x  
  
myclass = MyNumbers()  
myiter = iter(myclass)  
print(next(myiter))  
print(next(myiter))  
print(next(myiter))  
print(next(myiter))  
print(next(myiter))
```

Try it Yourself "

StopIteration

The example above would continue forever if you had enough `next()` statements, or if it was used in a for loop.

To prevent the iteration to go on forever, we can use the `StopIteration` statement.

In the `__next__()` method, we can add a terminating condition to raise an error if the iteration is done a specified number of times:

Example

Stop after 20 iterations:

```
class MyNumbers:  
    def __iter__(self):  
        self.a = 1  
        return self  
    def __next__(self):  
        if self.a <= 20:  
            x = self.a  
            self.a += 1  
            return x  
        else:  
            raise StopIteration  
myclass = MyNumbers()  
myiter = iter(myclass)  
for x in myiter:  
    print(x)
```

Python Math

Python has a set of built-in math functions, including an extensive math module, that allows you to perform mathematical tasks on numbers.

Built-in Math Functions

The `min()` and `max()` functions can be used to find the lowest or highest value in an iterable:

Example

```
x = min(5, 10, 25)  
y = max(5, 10, 25)  
print(x)  
print(y)
```

Try it Yourself "

The `abs()` function returns the absolute (positive) value of the specified number:

Example

```
x = abs(-7.25)  
print(x)
```

Try it Yourself "

The `pow(x, y)` function returns the value of `x` to the power of `y` (xy).

Example

Return the value of 4 to the power of 3 (same as $4 * 4 * 4$):

```
x = pow(4, 3)
print(x)
```

Try it Yourself "

The Math Module

Python has also a built-in module called `math`, which extends the list of mathematical functions.

To use it, you must import the `math` module:

import math

When you have imported the `math` module, you can start using methods and constants of the module.

The `math.sqrt()` method for example, returns the square root of a number:

Example

```
import math
x = math.sqrt(64)
print(x)
```

Try it Yourself "

The `math.ceil()` method rounds a number upwards to its nearest integer, and the `math.floor()` method rounds a number downwards to its nearest integer, and returns the result:

Example

```
import math
x = math.ceil(1.4)
y = math.floor(1.4)
print(x) # returns 2
print(y) # returns 1
```

Try it Yourself "

The `math.pi` constant, returns the value of PI (3.14...):

Example

```
import math
x = math.pi
print(x)
```

Complete Math Module Reference

In our Math Module Reference you will find a complete reference of all methods and constants that belongs to the `Math` module.

Python User Input

User Input

Python allows for user input.

That means we are able to ask the user for input.

The method is a bit different in Python 3.6 than Python 2.7.

Python 3.6 uses the `input()` method.

Python 2.7 uses the `raw_input()` method.

The following example asks for the username, and when you entered the username, it gets printed on the screen:

Python 3.6

```
username = input("Enter username:")
print("Username is: " + username)
```

Run Example "

Python 2.7

```
username = raw_input("Enter username:")
print("Username is: " + username)
```

Run Example "

Python stops executing when it comes to the `input()` function, and continues when the user has given some input.

Object Oriented Programming and JAVA Language

Objectives: At the end of this lesson you shall be able to

- what is JAVA
- concepts of OOPs.

JavaTutorial

What is Java

Our core Java programming tutorial is designed for students and working professionals. Java is an object-oriented, class-based, concurrent, secured and general-purpose computer-programming language. It is a widely used robust technology.

Java is a programming language and a platform. Java is a high level, robust, object-oriented and secure programming language.

Java was developed by Sun Microsystems (which is now the subsidiary of Oracle) in the year 1995. James Gosling is known as the father of Java. Before Java, its name was Oak. Since Oak was already a registered company, so James Gosling and his team changed the name from Oak to Java.

Platform: Any hardware or software environment in which a program runs, is known as a platform. Since Java has a runtime environment (JRE) and API, it is called a platform.

JavaExample

Let's have a quick look at Java programming example. A detailed description of Hello Java example is available in next page.

Simple.java

```
class Simple{
    public static void main
        (String args[])
    {
        System.out.println("Hello Java");
    }
}
```

Application

According to Sun, 3 billion devices run Java. There are many devices where Java is currently used. Some of them areas follows:

- 1 Desktop Applications such as acrobat reader, media player, antivirus, etc.
- 2 Web Applications such as irctc.co.in, javatpoint.com, etc.
- 3 Enterprise Applicationssuch as banking applications.
- 4 Mobile
- 5 EmbeddedSystem
- 6 SmartCard

7 Robotics

8 Games,etc.

Types of Java Applications

There are mainly 4 types of applications that can be created using Java programming:

1 Standalone Application

Standalone applications are also known as desktop applications or window-based applications. These are traditional software that we need to install on every machine. Examples of standalone application are Media Player, antivirus, etc. AWT and Swing are used in Java for creating stand alone applications.

2 Web Application

An application that runs on the server side and creates a dynamic page is called a web application. Currently, Servlet, JSP, Struts, Spring, Hibernate, JSF, etc. technologies are used for creating web applications in Java.

3 Enterprise Application

An application that is distributed in nature, such as banking applications, etc. is called an enterprise application. It has advantages like high-level security, load balancing, and clustering. In Java, EJB is used for creating enterprise applications.

4 Mobile Application

An application which is created for mobile devices is called a mobile application. Currently, Android and Java ME are used for creating mobile applications.

Java Platforms/Editions

There are 4 platform editions of Java:

1 JavaSE (Java Standard Edition)

It is a Java programming platform. It includes Java programming APIs such as java.lang, java.io, java.net, java.util, java.sql, java.math etc. It includes core topics like OOPs, String, Regex, Exception, Inner classes, Multi threading, I/O Stream, Networking, AWT, Swing, Reflection, Collection, etc.

2 JavaEE (Java Enterprise Edition)

It is an enterprise platform that is mainly used to develop web and enterprise applications. It is built on top of the Java SE platform. It includes topics like Servlet, JSP, Web Services, EJB, JPA, etc.

3 JavaME (Java Micro Edition)

It is a micro platform that is dedicated to mobile applications.

4 JavaFX

It is used to develop rich internet applications. It uses a light weight user interface API.

Prerequisite

To learn Java, you must have the basic knowledge of C/C++ programming language.

Audience

Our Java programming tutorial is designed to help beginners and professionals.

History of Java

The history of Java is very interesting. Java was originally designed for interactive television, but it was too advanced technology for the digital cable television industry at the time. The history of Java starts with the Green Team. Java team members (also known as Green Team), initiated this project to develop a language for digital devices such as set-top boxes, televisions, etc. However, it was best suited for internet programming. Later, Java technology was incorporated by Netscape.

The principles for creating Java programming were "Simple, Robust, Portable, Platform-independent, Secured, High Performance, Multithreaded, Architecture Neutral, Object-Oriented, Interpreted, and Dynamic". Java was developed by James Gosling, who is known as the father of Java, in 1995. James Gosling and his team members started the project in the early '90s.

Currently, Java is used in internet programming, mobile devices, games, e-business solutions, etc. Following are given significant points that describe the history of Java.

James Gosling-founder of java



- 1 James Gosling, Mike Sheridan, and Patrick Naughton initiated the Java language project in June 1991. The small team of sun engineers called Green Team.
- 2 Initially it was designed for small, embedded systems in electronic appliances like set-top boxes.
- 3 Firstly, it was called "Greentalk" by James Gosling, and the file extension was .gt.

- 4 After that, it was called Oak and was developed as a part of the Green project.

Why Java was named as "Oak"? Java History from Oak to Java

- 5 Why Oak? Oak is a symbol of strength and chosen as a national tree of many countries like the U.S.A., France, Germany, Romania, etc.
- 6 In 1995, Oak was renamed as "Java" because it was already a trade mark by Oak Technologies.

Why Java Programming named "Java"?

- 7 Why had they chose the name Java for Java language? The team gathered to choose a new name. The suggested words were "dynamic", "revolutionary", "Silk", "jolt", "DNA", etc. They wanted something that reflected the essence of the technology: revolutionary, dynamic, lively, cool, unique, and easy to spell, and fun to say.

According to James Gosling, "Java was one of the top choices along with Silk". Since Java was so unique, most of the team members preferred Java than other names.

- 8 Java is an island in Indonesia where the first coffee was produced (called Java coffee). It is a kind of espresso bean. Java name was chosen by James Gosling while having a cup of coffee near his office.
- 9 Notice that Java is just a name, not an acronym.
- 10 Initially developed by James Gosling at Sun Microsystems (which is now a subsidiary of Oracle Corporation) and released in 1995.
- 11 In 1995, Time magazine called Java one of the Ten Best Products of 1995.
- 12 JDK 1.0 was released on January 23, 1996. After the first release of Java, there have been many additional features added to the language. Now Java is being used in Windows applications, Web applications, enterprise applications, mobile applications, cards, etc. Each new version adds new features in Java.

Java Version History

Many Java versions have been released till now. The current stable release of Java is Java SE 18.

- 1 JDK Alpha and Beta (1995)
- 2 JDK 1.0 (23rd Jan 1996)
- 3 JDK 1.1 (19th Feb 1997)
- 4 J2SE 1.2 (8th Dec 1998)
- 5 J2SE 1.3 (8th May 2000)
- 6 J2SE 1.4 (6th Feb 2002)
- 7 J2SE 5.0 (30th Sep 2004)
- 8 Java SE 6 (11th Dec 2006)
- 9 Java SE 7 (28th July 2011)
- 10 Java SE 8 (18th Mar 2014)

11 JavaSE 9 (21stSep2017)

12 JavaSE 10 (20thMar2018)

13 JavaSE11 (September2018)

14 JavaSE12(March2019)

15 JavaSE13(September2019)

16 JavaSE14(Mar2020)

17 JavaSE15(September2020)

18 JavaSE16(Mar2021)

19 JavaSE17(September2021)

20 JavaSE 18 (tobe releasedbyMarch2022)

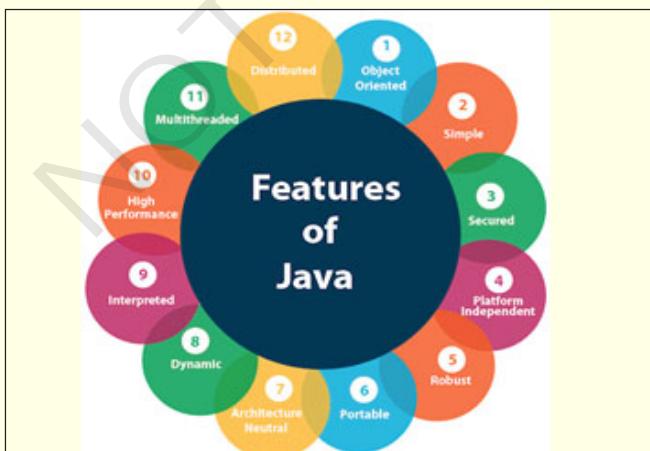
Since Java SE 8 release, the Oracle corporation follows a pattern in which every even version is release in Marchmonthandan oddversionreleased inSeptember month.

Features of Java

The primary objective of Java programming language creation was to make it portable, simple and secure programming language. Apart from this, there are also some excellent features which play an important role in the popularity of this language. The features of Java are also known as Java buzzwords.

A list of the most important features of the Java language is given below.

- 1 Simple
- 2 Object-Oriented
- 3 Portable
- 4 Platform independent
- 5 Secured
- 6 Robust
- 7 Architecture neutral
- 8 Interpreted
- 9 High Performance
- 10 Multithreaded
- 11 Distributed
- 12 Dynamic



Simple

Java is very easy to learn, and its syntax is simple, clean and easy to understand. According to Sun Microsystem, Java language is a simple programming language because:

- Java syntax is based on C++ (so easier for programmers to learn it after C++).
- Java has removed many complicated and rarely-used features, for example, explicit pointers, operator overloading, etc.
- There is no need to remove unreferenced objects because there is an Automatic Garbage Collection in Java.

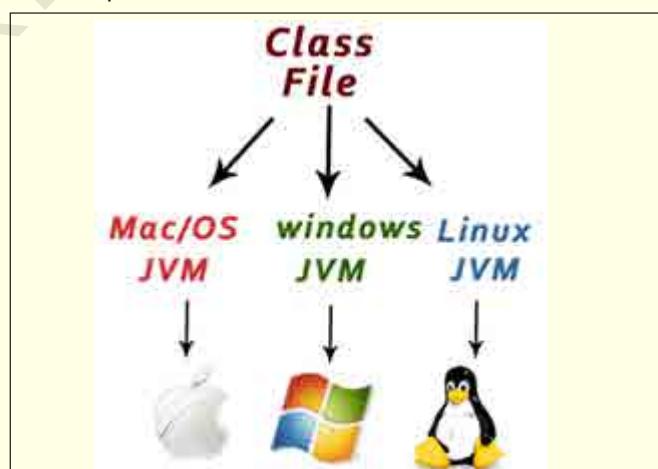
Object-oriented

Java is an object-oriented programming language. Everything in Java is an object. Object-oriented means we organize our software as a combination of different types of objects that incorporate both data and behavior.

Object-oriented programming (OOPs) is a methodology that simplifies software development and maintenance by providing some rules.

Basic concepts of OOPs are:

- 1 Object
- 2 Class
- 3 Inheritance
- 4 Polymorphism
- 5 Abstraction
- 6 Encapsulation



Platform Independent

Java is platform independent because it is different from other languages like C, C++, etc. which are compiled into platform specific machines while Java is a write once, run anywhere language. A platform is the hardware or software environment in which a program runs.

There are two types of platforms software-based and hardware-based. Java provides a software-based platform.

The Java platform differs from most other platforms in the sense that it is a software-based platform that runs on top of other hardware-based platforms. It has two components:

1 Runtime Environment

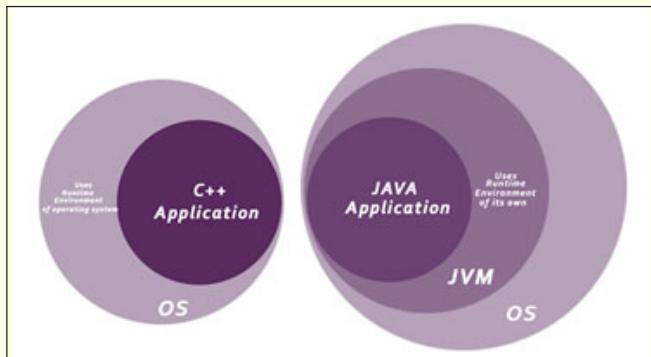
2 API (Application Programming Interface)

Java code can be executed on multiple platforms, for example, Windows, Linux, Sun Solaris, Mac/OS, etc. Java code is compiled by the compiler and converted into bytecode. This bytecode is a platform-independent code because it can be run on multiple platforms, i.e., Write Once and Run Anywhere (WORA).

Secured

Java is best known for its security. With Java, we can develop virus-free systems. Java is secured because:

- No explicit pointer
- Java Programs run inside a virtual machine sandbox



- **Classloader:** Classloader in Java is a part of the Java Runtime Environment (JRE) which is used to load Java classes into the Java Virtual Machine dynamically. It adds security by separating the package for the classes of the local file system from those that are imported from network sources.
- **Bytecode Verifier:** It checks the code fragments for illegal code that can violate access rights to objects.
- **Security Manager:** It determines what resources a class can access such as reading and writing to the local disk.

Java language provides these securities by default. Some security can also be provided by an application developer explicitly through SSL, JAAS, Cryptography, etc.

Robust

The English meaning of Robust is strong. Java is robust because:

- It uses strong memory management.
- There is a lack of pointers that avoids security problems.

- Java provides automatic garbage collection which runs on the Java Virtual Machine to get rid of objects which are not being used by a Java application anymore.
- There are exception handling and the type checking mechanism in Java. All these points make Java robust.

Architecture-neutral

Java is architecture neutral because there are no implementation dependent features, for example, the size of primitive types is fixed.

In C programming, int data type occupies 2 bytes of memory for 32-bit architecture and 4 bytes of memory for 64-bit architecture. However, it occupies 4 bytes of memory for both 32 and 64-bit architectures in Java.

Portable

Java is portable because it facilitates you to carry the Java bytecode to any platform. It doesn't require any implementation.

High-performance

Java is faster than other traditional interpreted programming languages because Java bytecode is "close" to native code. It is still a little bit slower than a compiled language (e.g., C++). Java is an interpreted language that is why it is slower than compiled languages, e.g., C, C++, etc.

Distributed

Java is distributed because it facilitates users to create distributed applications in Java. RMI and EJB are used for creating distributed applications. This feature of Java makes us able to access files by calling the methods from any machine on the internet.

Multi-threaded

A thread is like a separate program, executing concurrently. We can write Java programs that deal with many tasks at once by defining multiple threads. The main advantage of multi-threading is that it doesn't occupy memory for each thread. It shares a common memory area. Threads are important for multi-media, Web applications, etc.

Dynamic

Java is a dynamic language. It supports the dynamic loading of classes. It means classes are loaded on demand. It also supports functions from its native languages, i.e., C and C++.

Java supports dynamic compilation and automatic memory management (garbage collection).

Demonstrate writing JAVA programs

Objectives: At the end of this lesson you shall be able to

- java Virtual Machine
- internal Architecture of JVM.

JVM (Java Virtual Machine) Architecture

JVM (Java Virtual Machine) is an abstract machine. It is a specification that provides runtime environment in which java bytecode can be executed.

JVMs are available for many hardware and software platforms (i.e. JVM is platform dependent).

What is JVM

It is:

- 1 **A specification** where working of Java Virtual Machine is specified. But implementation provider is independent to choose the algorithm. Its implementation has been provided by Oracle and other companies.
- 2 **An implementation** Its implementation is known as JRE (Java Runtime Environment).
- 3 **Runtime Instance** Whenever you write java command on the command prompt to run the java class, an instance of JVM is created.

What it does

The JVM performs following operation:

- Loads code
- Verifies code
- Executes code
- Provides runtime environment

JVM provides definitions for the:

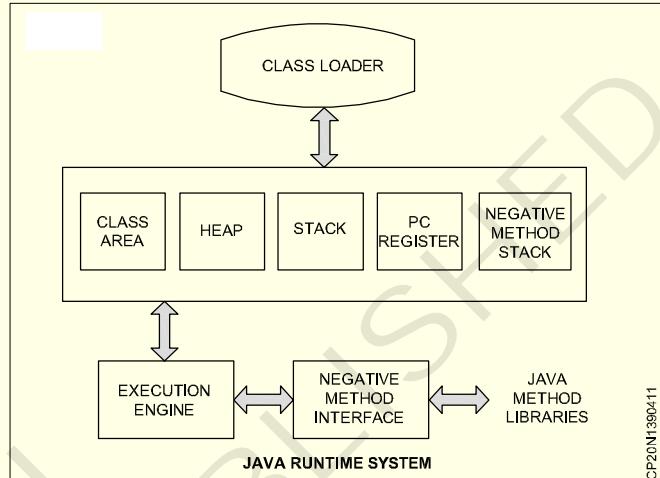
- Memory area
- Class file format
- Register set
- Garbage-collected heap
- Fatal error reporting etc.

JVM Architecture

Let's understand the internal architecture of JVM. It contains classloader, memory area, execution engine etc.

1 Classloader

Classloader is a subsystem of JVM which is used to load class files. Whenever we run the java program, it is loaded first by the classloader. There are three built-in classloaders in Java.



- 1 **Bootstrap ClassLoader:** This is the first classloader which is the super class of Extension classloader. It loads the rt.jar file which contains all class files of Java Standard Edition like java.lang package classes, java.net package classes, java.util package classes, java.io package classes, java.sql package classes etc.
- 2 **Extension ClassLoader:** This is the child classloader of Bootstrap and parent classloader of System classloader. It loads the jar files located inside \$JAVA_HOME/jre/lib/ext directory.
- 3 **System/Application ClassLoader:** This is the child classloader of Extension classloader. It loads the classfiles from classpath. By default, classpath is set to current directory. You can change the classpath using "-cp" or "-classpath" switch. It is also known as Application classloader.

```

1 //Let's see an example to print the classloader name
2 public class ClassLoaderExample
3 {
4     public static void main(String[] args)
5     {
6         // Let's print the classloader name of current class.
7         //Application/System classloader will load this class
8         Class c=ClassLoaderExample.class;
9         System.out.println(c.getClassLoader());
10 //If we print the classloader name of String, it will print
11 //null because it is an
12 //in-built class which is found in rt.jar, so it is loaded by
13 //Bootstrap classloader
    
```

```
12 System.out.println (String.class.getClassLoader());
13 }
14 }
```

Output:

These are the internal classloaders provided by Java. If you want to create your own classloader, you need to extend the ClassLoader class.

Class (Method) Area

Class(Method) Area stores per-class structures such as the runtime constant pool, field and method data, the code for methods.

Heap

It is the runtime data area in which objects are allocated.

Stack

Java Stack stores frames. It holds local variables and partial results, and plays a part in method invocation and return.

Each thread has a private JVM stack, created at the same time as thread.

A new frame is created each time a method is invoked. A frame is destroyed when its method invocation completes.

Program Counter Register

PC (program counter) register contains the address of the Java virtual machine instruction currently being executed.

Native Method Stack

It contains all the native methods used in the application.

Execution Engine

It contains:

- 1 A virtual processor
- 2 **Interpreter:** Read bytecode stream then execute the instructions.
- 3 **Just-In-Time(JIT) compiler:** It is used to improve the performance. JIT compiles parts of the byte code that have similar functionality at the same time, and hence reduces the amount of time needed for compilation. Here, the term "compiler" refers to a translator from the instruction set of a Java virtual machine (JVM) to the instruction set of a specific CPU.

Java Native Interface: Java Native Interface (JNI) is a framework which provides an interface to communicate with another application written in another language like C, C++, Assembly etc. Java uses JNI framework to send output to the Console or interact with OS libraries.

What is Bytecode in Java

Overview

Bytecode in Java is a set of instructions for the Java Virtual Machine that is responsible for interpreting them. Upon compiling a Java program, Java bytecode is generated that can be executed on any platform (platform-independent) using the JVM. It is present in a .class file.

Scope

This article aims to:

- Explain the concept of bytecode in Java.
- Help understand the advantages and disadvantages of bytecode in Java.

What is Bytecode in Java?

Can you imagine a world where we have to write code multiple times for each device with varying specifications like operating system, processor architecture, etc.? It will be a tedious process. Keeping these things in mind, Java developers came up with the concept of Write Once Read Anywhere, which got incorporated as a feature in the Java language. To achieve this, a code known as bytecode gets generated after compiling our Java source code. We will now discuss the same.

Bytecode in Java is a set of instructions for the Java Virtual Machine. Java Virtual Machine, abbreviated as JVM, enables a computer to run code written in Java. When a Java program is compiled, the bytecode gets generated. It is equivalent to the assembler in C++.

Bytecode is a platform-independent set of instructions primarily because it is interpreted and executed by the JVM. Code that runs on multiple computer architectures without any modification is called machine or platform-independent code.

Bytecode lies in-between low-level and high-level sets of instructions. Since we write our code in a high-level language, it gets compiled into bytecode, and later, JVM interprets it into machine code, a low-level set of instructions ready to be executed. Hence, we call bytecode a code between low-level and high-level language.

Fact:

Java is a compiled and interpreted language, unlike most other languages, which are either compiled or interpreted.

How does it work?

Let us understand the above process with an explanation:

- The code we write in Java is called the source code, which is written in a high-level language. A high-level language is a programmer-friendly language with statements written in English and is closer to human languages. The extension of the Java file is ".java".
- When we compile the program, the compiler compiles the ".java" file and generates a ".class" file. It contains the bytecode.
- The bytecode allows us to run the ".class" file on any other platform.
- But this bytecode requires an interpreter to execute it. Here the JVM comes into the picture. JVM has an interpreter. It executes the code piece by piece, i.e., one statement at a time, until it finds an error or is done with executing the end of the code.

Hence, bytecode is said to be between low-level language and high-level language.

Let us look at an example-

Suppose you have written the code in Java for a calculator app. This is how we can easily understand the way it gets processed.

Advantages

Platform Independence

Java project was started with the purpose of running the same piece of code on any device without any modifications. This concept of bytecode in Java helps in achieving this goal. Bytecodes can be different for different systems, but they are interpreted and executed by the JVM on all systems alike.

Bytecode is essentially the machine-level code that runs on the Java Virtual Machine. Whenever a class is loaded, it gets a stream of bytecode per class method. Whenever that method is called during the execution of a program, the bytecode for that method gets invoked. Javac not only compiles the program but also generates the bytecode for the program. Thus, we have realized that the bytecode implementation makes Java a platform-independent language.

Portability

It helps to add portability which ensures that the code can run on a variety of devices. This resonates well with the saying "write once, run anywhere" principle implemented by Java. We also don't need to write the code again on any other device.

Disadvantages

- The bytecode cannot run without an interpreter or JVM. If any device doesn't have JVM, bytecode won't run on that device.
- It is difficult to analyze the bytecode as it is in the form of binary and not understandable by humans.

Conclusion

- Bytecode in Java is a set of instructions for the Java Virtual Machine.
- Bytecode is a platform-independent code.
- Bytecode is a code that lies between low-level language and high-level language.
- After the Java code is compiled, the bytecode gets generated, which can be executed on any machine using JVM.
- When we compile the program, the compiler compiles the ".java" file and generates a ".class" file.
- The bytecode needs an interpreter to run it; hence, JVM acts as an interpreter.
- Java is both compiled and interpreted.

How to Set Classpath in Java?

- Read
- Discuss
- Practice

- Video

- Courses

CLASSPATH describes the location where all the required files are available which are used in the application. Java Compiler and JVM (Java Virtual Machine) use CLASSPATH to locate the required files. If the CLASSPATH is not set, Java Compiler will not be able to find the required files and hence will throw the following error.

Error: Could not find or load main class <class name> (e.g. GFG)

The above error is resolved when CLASSPATH is set.

Java

```
// If the following code is run when the CLASSPATH is not
// set, it will throw the above error.
// If it is set, we get the desired result
import java.io.*;
class GFG {
    public static void main(String[] args)
    {
        // prints GeeksForGeeks to the console
        System.out.println("GeekForGeeks!");
    }
}
```

Output

GeekForGeeks!

Set the CLASSPATH in JAVA in Windows

Command Prompt:

```
set PATH=.;C:\Program Files\Java\JDK1.6.20\bin
```

Note: Semi-colon (;) is used as a separator and dot (.) is the default value of CLASSPATH in the above command.

GUI:

- 1 Select Start
- 2 Go to the Control Panel
- PICTURE
- 3 Select System and Security
- PICTURE
4. Select Advanced System settings
- PICTURE
- 5 Click on Environment Variables
- PICTURE
- 6 Click on New under System Variables
- PICTURE

- 7 Add CLASSPATH as variable name and path of files as a variable value.

PICTURE

Select OK.

Set the CLASSPATH on Linux

Command Line:

Find out where you have installed Java, basically, it's in /usr/lib/jvm path. Set the CLASSPATH in /etc/environment using

```
sudo <editor name> /etc/environment
```

Add the following lines,

```
JAVA_HOME = "/usr/lib/jvm/<java folder (eg. java-1.8.0-openjdk-amd64>)/bin"
```

```
export JAVA_HOME
```

```
CLASSPATH=".:/usr/lib/jvm/<java folder>/lib:/home/name/Desktop"
```

```
export CLASSPATH
```

Note: Colon (:) is used as a separate directory and dot (.) is the default value of CLASSPATH in the above command.

To check the current CLASSPATH, run

```
echo ${CLASSPATH}
```

Keywords

JavaKeywords

Java keywords are also known as reserved words. Keywords are particular words that act as a key to a code. These are predefined words by Java so they cannot be used as a variable or object name or class name.

List of Java Keywords

A list of Java keywords or reserved words are given below:

- 1 **abstract:** Java abstract keyword is used to declare an abstract class. An abstract class can provide the implementation of the interface. It can have abstract and non-abstract methods.
- 2 **boolean:** Java boolean keyword is used to declare a variable as a boolean type. It can hold True and False values only.
- 3 **break:** Java break keyword is used to break the loop or switch statement. It breaks the current flow of the program at specified conditions.
- 4 **byte:** Java byte keyword is used to declare a variable that can hold 8-bit data values.
- 5 **case:** Java case keyword is used with the switch statements to mark blocks of text.
- 6 **catch:** Java catch keyword is used to catch the exceptions generated by try statements. It must be used after the try block only.
- 7 **char:** Java char keyword is used to declare a variable that can hold unsigned 16-bit Unicode characters

- 8 **class:** Java class keyword is used to declare a class.

- 9 **continue:** Java continue keyword is used to continue the loop. It continues the current flow of the program and skips the remaining code at the specified condition.

- 10 **default:** Java default keyword is used to specify the default block of code in a switch statement.

- 11 **do:** Java do keyword is used in the control statement to declare a loop. It can iterate a part of the program several times.

- 12 **double:** Java double keyword is used to declare a variable that can hold 64-bit floating-point number.

- 13 **else:** Java else keyword is used to indicate the alternative branches in an if statement.

- 14 **enum:** Java enum keyword is used to define a fixed set of constants. Enum constructors are always private or default.

- 15 **extends:** Java extends keyword is used to indicate that a class is derived from another class or interface.

- 16 **final:** Java final keyword is used to indicate that a variable holds a constant value. It is used with a variable. It is used to restrict the user from updating the value of the variable.

- 17 **finally:** Java finally keyword indicates a block of code in a try-catch structure. This block is always executed whether an exception is handled or not.

- 18 **float:** Java float keyword is used to declare a variable that can hold a 32-bit floating-point number.

- 19 **for:** Java for keyword is used to start a for loop. It is used to execute a set of instructions/functions repeatedly when some condition becomes true. If the number of iteration is fixed, it is recommended to use for loop.

- 20 **if:** Java if keyword tests the condition. It executes the if block if the condition is true.

- 21 **implements:** Java implements keyword is used to implement an interface.

- 22 **import:** Java import keyword makes classes and interfaces available and accessible to the current source code.

- 23 **instance of:** Java instanceof keyword is used to test whether the object is an instance of the specified class or implements an interface.

- 24 **int:** Java int keyword is used to declare a variable that can hold a 32-bit signed integer.

- 25 **interface:** Java interface keyword is used to declare an interface. It can have only abstract methods.

- 26 **long:** Java long keyword is used to declare a variable that can hold a 64-bit integer.

- 27 **native:** Java native keyword is used to specify that a method is implemented in native code using JNI (Java Native Interface).

- 28 **new**: Java new keyword is used to create new objects.
- 29 **null**: Java null keyword is used to indicate that a reference does not refer to anything. It removes the garbage value.
- 30 **package**: Java package keyword is used to declare a Java package that includes the classes.
- 31 **private**: Java private keyword is an access modifier. It is used to indicate that a method or variable maybe accessed only in the class in which it is declared.
- 32 **protected**: Java protected keyword is an access modifier. It can be accessible within the package and outside the package but through inheritance only. It can't be applied with the class.
- 33 **public**: Java public keyword is an access modifier. It is used to indicate that an item is accessible anywhere. It has the widest scope among all other modifiers.
34. **return**: Java return keyword is used to return from a method when its execution is complete.
- 35 **short**: Java short keyword is used to declare a variable that can hold a 16-bit integer.
- 36 **static**: Java static keyword is used to indicate that a variable or method is a class method. The static keyword in Java is mainly used for memory management.
- 37 **strictfp**: Java strictfp is used to restrict floating-point calculations to ensure reportability.
- 38 **super**: Java super keyword is a reference variable that is used to refer to parent class objects. It can be used to invoke the immediate parent class method.
- 39 **switch**: The Java switch keyword contains a switch statement that executes code based on test value. The switch statement tests the equality of a variable against multiple values.
- 40 **synchronized**: Java synchronized keyword is used to specify the critical sections or methods in multi-threaded code.
- 41 **this**: Java this keyword can be used to refer the current object in a method or constructor.
- 42 **throw**: The Java throw keyword is used to explicitly throw an exception. The throw keyword is mainly used to throw custom exceptions. It is followed by an instance.
- 43 **throws**: The Java throws keyword is used to declare an exception. Checked exceptions can be propagated with throws.
- 44 **transient**: Java transient keyword is used in serialization. If you define any data member as transient, it will not be serialized.
- 45 **try**: Java try keyword is used to start a block of code that will be tested for exceptions. The try block must be followed by either catch or finally block.
- 46 **void**: Java void keyword is used to specify that a method does not have a return value.
47. **volatile**: Java volatile keyword is used to indicate that a variable may change asynchronously.
- 48 **while**: Java while keyword is used to start a while loop. This loop iterates a part of the program several times. If the number of iterations is not fixed, it is recommended to use the while loop.

Datatypes in Java

Objectives: At the end of this lesson you shall be able to

- **datatypes type**
- **javaPrimitiveDataTypes**
- **non-primitivedatatypes**

DataTypes in Java

Data types specify the different sizes and values that can be stored in the variable. There are two types of datatypes in Java:

Primitive data types: The primitive data types include boolean, char, byte, short, int, long, float and double.

Non-primitive datatypes: The non-primitive data types include Classes, Interfaces, and Arrays.

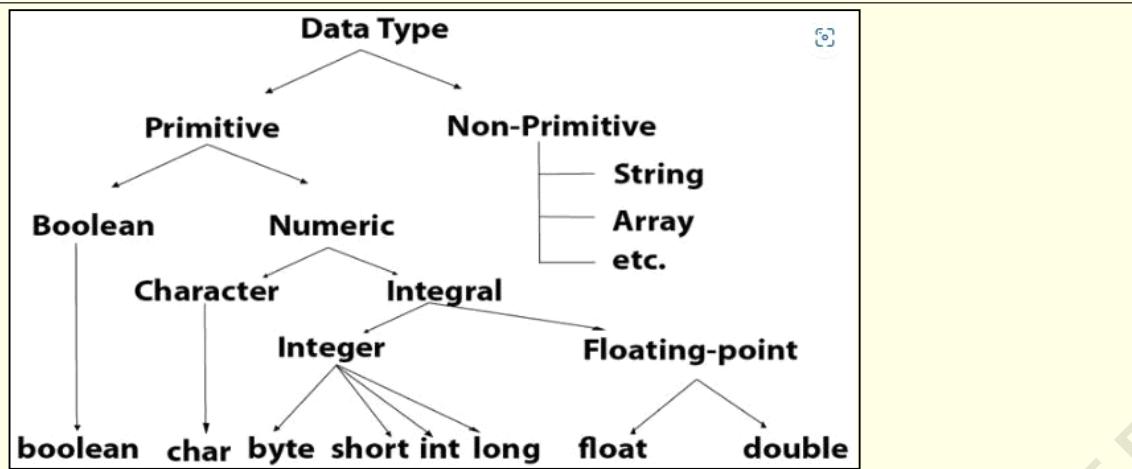
Java Primitive Data Types

In Java language, primitive data types are the building blocks of data manipulation. These are the most basic datatypes available in Java language.

Note: Java is a statically-typed programming language. It means, all variables must be declared before its use. That is why we need to declare variable's type and name.

There are 8 types of primitive datatypes:

- 1 boolean datatype
- 2 byte datatype
- 3 char datatype
- 4 short datatype
- 5 int datatype
- 6 long datatype
- 7 float datatype
- 8 double datatype



Java DataTypes

DataType	Default Value	Defaultsize
boolean	false	1bit
char	'\u0000'	2byte
byte	0	1byte
short	0	2byte
int	0	4byte
long	0L	8byte
float	0.0f	4byte
double	0.0d	8byte

BooleanDataType

The Boolean data type is used to store only two possible values: true and false. This data type is used for simple flags that track true/false conditions.

The boolean data type specifies one bit of information, but its "size" can't be defined precisely.

Example:

Booleanone =false

ByteDataType

The byte data type is an example of primitive data type. It is an 8-bit signed two's complement integer. Its value-range lies between -128 to 127 (inclusive). Its minimum value is -128 and maximum value is 127. Its default value is 0.

The byte data type is used to save memory in large arrays where the memory savings is most required. It saves space because a byte is 4 times smaller than an integer. It can also be used in place of "int" datatype.

Example:

byte a = 10,byteb= -20

ShortDataType

The short data type is a 16-bit signed two's complement integer. Its value-range lies between -32,768 to 32,767 (inclusive). Its minimum value is -32,768 and maximum value is 32,767. Its default value is 0.

The short data type can also be used to save memory just like byte data type. A short data type is 2 times smaller than an integer.

Example:

shorts =10000,shortr=-5000

IntDataType

The int data type is a 32-bit signed two's complement integer. Its value-range lies between -2,147,483,648 (-2^31) to 2,147,483,647 (2^31 -1) (inclusive). Its minimum value is -2,147,483,648 and maximum value is 2,147,483,647. Its default value is 0.

The int data type is generally used as a default data type for integral values unless if there is no problem about memory.

Example:

inta= 100000,intb =-200000

LongDataType

The long data type is a 64-bit two's complement integer. Its value-range lies between -9,223,372,036,854,775,808 (-2^63) to 9,223,372,036,854,775,807 (2^63 -1) (inclusive). Its minimum value is -9,223,372,036,854,775,808 and maximum value is 9,223,372,036,854,775,807. Its default value is 0. The longdatatype is used when you need a range of values more than those provided by int.

Example:

long a=100000L,long b=-200000L

FloatDataType

The float data type is a single-precision 32-bit IEEE 754 floating point. Its value range is unlimited. It is recommended to use a float (instead of double) if you need to save memory in large arrays of floating point numbers. The float data type should never be used for precise values, such as currency. Its default value is 0.0F.

Example:

floatf1 =234.5f

DoubleDataType

The double data type is a double-precision 64-bit IEEE 754 floating point. Its value range is unlimited. The double data type is generally used for decimal values just like float. The double data type also should never be used for precise values, such as currency. Its default value is 0.0d.

Example:

```
doubled1=12.3
```

CharDataType

The char data type is a single 16-bit Unicode character. Its value-range lies between '\u0000' (or 0) to '\uffff' (or 65,535 inclusive). The char data type is used to store characters.

Example:

```
charletterA='A'
```

Why char uses 2 byte in java and what is \u0000?

It is because java uses Unicode system not ASCII code system. The \u0000 is the lowest range of Unicode system. To get detail explanation about Unicode visit next page.

Operators in Java

Operator in Java is a symbol that is used to perform operations. For example: +, -, *, / etc. There are many types of operators in Java which are given below:

- UnaryOperator,
- ArithmeticOperator,
- ShiftOperator,
- RelationalOperator,
- BitwiseOperator,
- LogicalOperator,
- TernaryOperator and
- AssignmentOperator.

Java Operator Precedence

Operator Type	Category	Precedence
Unary	postfix	expr++expr--
prefix	++expr--expr+expr+expr~!	
Arithmetic	multiplicative	* /%
additive	+-	
Shift	shift	<<>>>>
Relational	comparison	<><=> instanceof
equality	== !=	
Bitwise	bitwiseAND	&
bitwiseexclusiveOR	^	
bitwiseinclusiveOR		
Logical	logicalAND	&&
logicalOR		
Ternary	ternary	? :
Assignment	assignment	=+= *==%=&=^= = <=>=>>=

Java Unary Operator

The Java unary operators require only one operand. Unary operators are used to perform various operations i.e.:

- incrementing/decrementing a value by one
- negating an expression
- inverting the value of a boolean

Java Unary Operator Example: ++ and --

```
public class OperatorExample{
```

```
    public static void main(String args[]){
```

```
        int x=10;
```

```
        System.out.println(x++);//10 (11)
```

```
System.out.println(++x);//12
```

```
System.out.println(x--);//12 (11)
```

```
System.out.println(--x);//10
```

```
}
```

Output:

```
10
```

```
12
```

```
12
```

```
10
```

```

JavaUnaryOperatorExample2:++and--
public class OperatorExample{
    public static void main(String args[]){
        int a=10;
        int b=10;
        System.out.println(a++ + ++a); //10+12=22
        System.out.println(b+++b++); //10+11=21
    }
}

```

Output:

```

22
21

```

JavaUnaryOperatorExample:~and!

```

public class OperatorExample{
    public static void main(String args[]){
        int a=10;
        int b=-10;
        boolean c=true;
        boolean d=false;
        System.out.println(~a); // -11 (minus of total positive value
        which starts from 0)
        System.out.println(~b); // 9 (positive of total minus,
        positive starts from 0)
        System.out.println(!c); // false (opposite of boolean value)
        System.out.println(!d); // true
    }
}

```

Output:

```

-11
9
false
true

```

JavaArithmeticOperators

Java arithmetic operators are used to perform addition, subtraction, multiplication, and division. They act as basic mathematical operations.

Java Arithmetic Operator Example

```

public class OperatorExample{
    public static void main(String args[]){
        int a=10;
        int b=5;
        System.out.println(a+b); //15
        System.out.println(a-b); //5
        System.out.println(a*b); //50
        System.out.println(a/b); //2
        System.out.println(a%b); //0
    }
}

```

Output:

```

15
5
50
2
0

```

JavaArithmeticOperatorExample:Expression

```

public class OperatorExample{
    public static void main(String args[]){
        System.out.println(10*10/5+3-1*4/2);
    }
}

```

Output:

```

21

```

JavaLeftShiftOperator

The Java left shift operator `<<` is used to shift all of the bits in a value to the left side of a specified number of times.

Java Left Shift Operator

```

Example public class OperatorExample{
    public static void main(String args[]){
        System.out.println(10<<2); //10*2^2=10*4=40
        System.out.println(10<<3); //10*2^3=10*8=80
        System.out.println(20<<2); //20*2^2=20*4=80
        System.out.println(15<<4); //15*2^4=15*16=240
    }
}

```

Output:

```

40
80
80
240

```

JavaRightShiftOperator

The Java right shift operator `>>` is used to move the value of the left operand to right by the number of bits specified by the right operand.

Java Right Shift Operator Example

```

public class OperatorExample{
    public static void main(String args[]){
        System.out.println(10>>2); //10/2^2=10/4=2
        System.out.println(20>>2); //20/2^2=20/4=5
        System.out.println(20>>3); //20/2^3=20/8=2
    }
}

```

Output:

```
2
5
2
```

JavaShiftOperatorExample:>>vs>>>

```
public class OperatorExample{
    public static void main(String args[]){
        //For positive number, >> and >>> works same
        System.out.println(20>>2);
        System.out.println(20>>>2);
        //For negative number, >>> changes parity bit (MSB) to 0
        System.out.println(-20>>2);
        System.out.println(-20>>>2);
    }
}
```

Output:

```
5
5
-5
1073741819
```

JavaANDOperatorExample:Logical&&andBitwise&

The logical **&&** operator doesn't check the second condition if the first condition is false. It checks the second condition only if the first one is true.

The bitwise **&** operator always checks both conditions whether first condition is true or false.

```
public class OperatorExample{
    public static void main(String args[]){
        int a=10; int b=5; int c=20;
        System.out.println(a<b&&a<c);//false && true = false
        System.out.println(a<b&a<c);//false&true=false
    }
}
```

Output:

```
false
false
```

JavaANDOperatorExample:Logical&&vsBitwise&

```
public class OperatorExample{
    public static void main(String args[]){
        int a=10;
        int b=5;
        int c=20;
```

```
System.out.println(a<b&&a++<c);//false && true = false
System.out.println(a);//10 because second condition is not checked
System.out.println(a<b&a++<c);//false && true = false
System.out.println(a);//11 because second condition is checked
}}
```

Output:

```
false10
false11
```

JavaOROperatorExample:Logical||andBitwise |

The logical **||** operator doesn't check the second condition if the first condition is true. It checks the second condition only if the first one is false.

The bitwise **|** operator always checks both conditions whether first condition is true or false.

```
public class OperatorExample{
    public static void main(String args[]){
        {
            int a=10;
            int b=5;
            int c=20;
            System.out.println(a>b||a<c);//true || true = true
            System.out.println(a>b|a<c);//true|true=true
            //|| vs |
            System.out.println(a>b||a++<c);//true || true = true
            System.out.println(a);//10 because second condition is not checked
            System.out.println(a>b|a++<c);//true | true = true
            System.out.println(a);//11 because second condition is checked
        }
    }
}
```

Output:

```
true
true
true
10
true
11
```

JavaTernaryOperator

Java Ternary operator is used as one line replacement for if-then-else statement and used a lot in Java programming. It is the only conditional operator which takes three operands.

Java Ternary Operator Example

```
public class Operator Example {  
    public static void main (String args[]){int a=2;  
    int b=5;  
    int min=(a<b)?a:b;  
    System.out.println(min);  
}
```

Output:

2

Another Example:

```
public class OperatorExample{  
    public static void main(String args[]){  
        int a=10;  
        int b=5;  
        int min=(a<b)?a:b;System.out.println(min);  
    }
```

Output:

5

Java Assignment Operator

Java assignment operator is one of the most common operators. It is used to assign the value on its right to the operand on its left.

Java Assignment Operator Example

```
public class OperatorExample{  
    public static void main(String args[]){  
        int a=10;  
        int b=20;  
        a+=4;//a=a+4 (a=10+4)  
        b-=4;//b=b-4 (b=20-4)  
        System.out.println(a);  
        System.out.println(b);  
    }
```

Output:

14

16

Java Assignment Operator Example

```
public class OperatorExample{  
    public static void main(String[] args){  
        int a=10;  
        a+=3;//10+3  
        System.out.println(a);
```

a=4;//13-4

```
System.out.println(a);  
a*=2;//9*2  
System.out.println(a);  
a/=2;//18/2  
System.out.println(a);  
}}
```

Output:

13

9

18

9

Java Assignment Operator Example: Adding short

```
public class OperatorExample{  
    public static void main(String args[]){  
        short a=10;  
        short b=10;  
        //a+=b;//a=a+b internally so fine  
        a=a+b;//Compile time error because 10+10=20 now  
        int  
        System.out.println(a);  
    }}
```

Output:

Compile time error

After typecast:

```
public class OperatorExample{  
    {  
        public static void main(String args[]){  
            short a=10;  
            short b=10;  
            a=(short)(a+b);//20 which is int now converted to  
            short  
            System.out.println(a);  
        }  
    }}
```

Output:

20

Java I/O Streams

In this tutorial, we will learn about Java input/output streams and their types.

In Java, streams are the sequence of data that are read from the source and written to the destination.

An input stream is used to read data from the source. And, an output stream is used to write data to the destination.

```
classHelloWorld{  
publicstaticvoidmain(String[] args){  
    System.out.println("Hello, World!");  
}  
}  
Run Code
```

For example, in our first Hello World example, we have used `System.out` to print a string. Here, the `System.out` is a type of output stream.

Similarly, there are input streams to take input.

We will learn about input streams and output streams in detail in the later tutorials.

Types of Streams

Depending upon the data a stream holds, it can be classified into:

- Byte Stream
- Character Stream

```
importjava.util.Scanner;// Import the Scanner class  
  
classMain{  
  
publicstaticvoidmain(String[] args){  
  
Scanner myObj =newScanner(System.in);// Create a Scanner object  
  
System.out.println("Enter username");  
  
  
String userName = myObj.nextLine();// Read user input  
  
System.out.println("Username is: "+ userName);// Output user input  
}  
}
```

Run Example

If you don't know what a package is, read our Java Packages Tutorial.

Byte Stream

Byte stream is used to read and write a single byte (8 bits) of data.

All byte stream classes are derived from base abstract classes called `InputStream` and `OutputStream`.

Character Stream

Character stream is used to read and write a single character of data.

All the character stream classes are derived from base abstract classes `Reader` and `Writer`.

Java User Input (Scanner)

Java User Input

The `Scanner` class is used to get user input, and it is found in the `java.util` package.

To use the `Scanner` class, create an object of the class and use any of the available methods found in the `Scanner` class documentation. In our example, we will use the `nextLine()` method, which is used to read Strings:

Example

Input Types

In the example above, we used the `nextLine()` method, which is used to read Strings. To read other types, look at the table below:

Method	Description
nextBoolean()	Reads a boolean value from the user
nextByte()	Reads a byte value from the user
nextDouble()	Reads a double value from the user
nextFloat()	Reads a float value from the user
nextInt()	Reads a int value from the user
nextLine()	Reads a String value from the user
nextLong()	Reads a long value from the user
nextShort()	Reads a short value from the user

In the example below, we use different methods to read data of various types:

Example

```
import java.util.Scanner;
class Main{
    public static void main(String[] args){
        Scanner myObj = new Scanner(System.in);
        System.out.println("Enter name, age and salary:");
        // String input
        String name = myObj.nextLine();
        // Numerical input
        int age = myObj.nextInt();
        double salary = myObj.nextDouble();
        // Output input by user
    }
}
```

Java Console class methods

Method	Description
Reader reader()	It is used to retrieve the reader object associated with the console
String readLine()	It is used to read a single line of text from the console.
String readLine(String fmt, Object... args)	It provides a formatted prompt then reads the single line of text from the console.
char[] readPassword()	It is used to read password that is not being displayed on the console.
char[] readPassword(String fmt, Object... args)	It provides a formatted prompt then reads the password that is not being displayed on the console.
Console format(String fmt, Object... args)	It is used to write a formatted string to the console output stream.
Console printf(String format, Object... args)	It is used to write a string to the console output stream.
PrintWriter writer()	It is used to retrieve the PrintWriter object associated with the console.
void flush()	It is used to flushes the console.

```
System.out.println("Name: " + name);
System.out.println("Age: " + age);
System.out.println("Salary: " + salary);
}
}
```

Run Example

Note: If you enter wrong input (e.g. text in a numerical input), you will get an exception/error message (like "InputMismatchException").

You can read more about exceptions and how to handle errors in the Exceptions chapter.

Java Console Class

The Java Console class is be used to get input from console. It provides methods to read texts and passwords.

If you read password using Console class, it will not be displayed to the user.

The java.io.Console class is attached with system console internally. The Console class is introduced since 1.5.

Let's see a simple example to read text from console.

- 1 String text=System.console().readLine();
- 2 System.out.println("Text is: "+text);

Java Console class declaration

Let's see the declaration for Java.io.Console class:

- 1 Public final class Console extends Object implements Flushable

How to get the object of Console

System class provides a static method console() that returns the singleton instance of Console class.

```
1 public static Console console(){}
```

Let's see the code to get the instance of Console class.

```
1 Console c=System.console();
```

Java Console Example

```
1 import java.io.Console;
2 class ReadStringTest{
3     public static void main(String args[]){
4         Console c=System.console();
5         System.out.println("Enter your name: ");
6         String n=c.readLine();
7         System.out.println("Welcome "+n);
8     }
9 }
```

Output

```
Enter your name: Nakul Jain
Welcome Nakul Jain
```

Java Console Example to read password

```
1 import java.io.Console;
2 class ReadPasswordTest{
3     public static void main(String args[]){
4         Console c=System.console();
5         System.out.println("Enter password: ");
6         char[] ch=c.readPassword();
7         String pass=String.valueOf(ch);//converting char array
    into string
8         System.out.println("Password is: "+pass);
9     }
10 }
```

Output

```
Enter password:
Password is: 123
```

JAVA Program Flow Control

Objectives: At the end of this lesson you shall be able to

- conditional statements
- loops
- arrays.

Java Conditions and If Statements

You already know that Java supports the usual logical conditions from mathematics:

- Less than: $a < b$
- Less than or equal to: $a \leq b$
- Greater than: $a > b$
- Greater than or equal to: $a \geq b$
- Equal to: $a == b$
- Not Equal to: $a != b$

You can use these conditions to perform different actions for different decisions.

Java has the following conditional statements:

- Use if to specify a block of code to be executed, if a specified condition is true
- Use else to specify a block of code to be executed, if the same condition is false
- Use else if to specify a new condition to test, if the first condition is false
- Use switch to specify many alternative blocks of code to be executed

The if Statement

Use the if statement to specify a block of Java code to be executed if a condition is true.

Syntax

```
if(condition){
    // block of code to be executed if the condition is true
}
```

Note that if is in lowercase letters. Uppercase letters (If or IF) will generate an error.

In the example below, we test two values to find out if 20 is greater than 18. If the condition is true, print some text:

Example

```
if(20>18){
    System.out.println("20 is greater than 18");
}
```

Try it Yourself

We can also test variables:

Example

```
int x =20;
int y =18;
if(x > y){
    System.out.println("x is greater than y");
}
```

Try it Yourself "

Example explained

In the example above we use two variables, x and y, to test whether x is greater than y (using the `>` operator). As x is 20, and y is 18, and we know that 20 is greater than 18, we print to the screen that "x is greater than y".

The else Statement

Use the else statement to specify a block of code to be executed if the condition is false.

Syntax

```
if(condition){
    // block of code to be executed if the condition is true
}else{
    // block of code to be executed if the condition is false
}
```

Example

```
int time =20;
if(time <18){
    System.out.println("Good day.");
}else{
    System.out.println("Good evening.");
}
// Outputs "Good evening."
```

Try it Yourself

Example explained

In the example above, time (20) is greater than 18, so the condition is false. Because of this, we move on to the else condition and print to the screen "Good evening". If the time was less than 18, the program would print "Good day".

The else if Statement

Use the else if statement to specify a new condition if the first condition is false.

Syntax

```
if(condition1){  
    // block of code to be executed if condition1 is true  
}  
elseif(condition2){  
    // block of code to be executed if the condition1 is  
    // false and condition2 is true  
}  
else{  
    // block of code to be executed if the condition1 is  
    // false and condition2 is false  
}
```

Example

```
int time =22;  
if(time <10){  
    System.out.println("Good morning.");  
}  
elseif(time <18){  
    System.out.println("Good day.");  
}  
else{  
    System.out.println("Good evening.");  
}  
// Outputs "Good evening."
```

Try it Yourself

Example explained

In the example above, time (22) is greater than 10, so the first condition is false. The next condition, in the else if statement, is also false, so we move on to the else condition since condition1 and condition2 is both false - and print to the screen "Good evening".

However, if the time was 14, our program would print "Good day."

Java Switch

Java Switch Statements

Instead of writing many if..else statements, you can use the switch statement.

The switch statement selects one of many code blocks to be executed:

Syntax

```
switch(expression){  
    case x:  
        // code block  
        break;  
    case y:  
        // code block  
        break;
```

```
// code block  
break;  
default:  
    // code block  
}
```

This is how it works:

- The switch expression is evaluated once.
- The value of the expression is compared with the values of each case.
- If there is a match, the associated block of code is executed.
- The break and default keywords are optional, and will be described later in this chapter

The example below uses the weekday number to calculate the weekday name:

Example

```
int day =4;  
switch(day){  
    case1:  
        System.out.println("Monday");  
        break;  
    case2:  
        System.out.println("Tuesday");  
        break;  
    case3:  
        System.out.println("Wednesday");  
        break;  
    case4:  
        System.out.println("Thursday");  
        break;  
    case5:  
        System.out.println("Friday");  
        break;  
    case6:  
        System.out.println("Saturday");  
        break;  
    case7:  
        System.out.println("Sunday");  
        break;  
}  
// Outputs "Thursday" (day 4)
```

Try it Yourself "

The break Keyword

When Java reaches a break keyword, it breaks out of the switch block.

This will stop the execution of more code and case testing inside the block.

When a match is found, and the job is done, it's time for a break. There is no need for more testing.

A break can save a lot of execution time because it "ignores" the execution of all the rest of the code in the switch block.

The default Keyword

The default keyword specifies some code to run if there is no case match:

Example

```
int day =4;  
switch(day){  
case6:  
    System.out.println("Today is Saturday");  
    break;  
case7:  
    System.out.println("Today is Sunday");  
    break;  
default:  
    System.out.println("Looking forward to the Weekend");  
}  
// Outputs "Looking forward to the Weekend"
```

Try it Yourself "

Note that if the default statement is used as the last statement in a switch block, it does not need a break.

Loops

Loops can execute a block of code as long as a specified condition is reached.

Loops are handy because they save time, reduce errors, and they make code more readable.

Java While Loop

The while loop loops through a block of code as long as a specified condition is true:

Syntax

```
while(condition){  
    // code block to be executed  
}
```

In the example below, the code in the loop will run, over and over again, as long as a variable (i) is less than 5:

Example

```
int i =0;  
while(i <5){  
    System.out.println(i);  
    i++;  
}
```

Try it Yourself "

Note: Do not forget to increase the variable used in the condition, otherwise the loop will never end!

The Do/While Loop

The do/while loop is a variant of the while loop. This loop will execute the code block once, before checking if the condition is true, then it will repeat the loop as long as the condition is true.

Syntax

```
do{  
    // code block to be executed  
}  
while(condition);
```

The example below uses a do/while loop. The loop will always be executed at least once, even if the condition is false, because the code block is executed before the condition is tested:

Example

```
int i =0;  
do{  
    System.out.println(i);  
    i++;  
}  
while(i <5);
```

Try it Yourself "

Do not forget to increase the variable used in the condition, otherwise the loop will never end!

Java For Loop

When you know exactly how many times you want to loop through a block of code, use the for loop instead of a while loop:

Syntax

```
for(statement 1; statement 2; statement 3){  
    // code block to be executed  
}
```

Statement 1 is executed (one time) before the execution of the code block.

Statement 2 defines the condition for executing the code block.

Statement 3 is executed (every time) after the code block has been executed.

The example below will print the numbers 0 to 4:

Example

```
for(int i =0; i <5; i++){  
    System.out.println(i);  
}
```

Try it Yourself "

Example explained

Statement 1 sets a variable before the loop starts (int i = 0).

Statement 2 defines the condition for the loop to run (i must be less than 5). If the condition is true, the loop will start over again, if it is false, the loop will end.

Statement 3 increases a value (i++) each time the code block in the loop has been executed.

Another Example

This example will only print even values between 0 and 10:

Example

```
for(int i =0; i <=10; i = i +2){  
    System.out.println(i);  
}
```

Try it Yourself

Nested Loops

It is also possible to place a loop inside another loop. This is called a nested loop.

The "inner loop" will be executed one time for each iteration of the "outer loop":

Example

```
// Outer loop  
for(int i =1; i <=2; i++){  
    System.out.println("Outer: "+ i);// Executes 2 times  
    // Inner loop  
    for(int j =1; j <=3; j++){  
        System.out.println(" Inner: "+ j);// Executes 6 times (2 * 3)  
    }  
}
```

Java Break

You have already seen the break statement used in an earlier chapter of this tutorial. It was used to "jump out" of a switch statement.

The break statement can also be used to jump out of a loop.

This example stops the loop when i is equal to 4:

Example

```
for(int i =0; i <10; i++){  
    if(i ==4){  
        break;  
    }  
    System.out.println(i);  
}
```

Try it Yourself

Java Continue

The continue statement breaks one iteration (in the loop), if a specified condition occurs, and continues with the next iteration in the loop.

This example skips the value of 4:

Example

```
for(int i =0; i <10; i++){  
    if(i ==4){  
        continue;  
    }  
    System.out.println(i);  
}
```

Try it Yourself

Break and Continue in While Loop

You can also use break and continue in while loops:

Break Example

```
int i =0;  
while(i <10){  
    System.out.println(i);  
    i++;  
    if(i ==4){  
        break;  
    }  
}
```

Try it Yourself "

Continue Example

```
int i =0;  
while(i <10){  
    if(i ==4){  
        i++;  
        continue;  
    }  
}
```

```
System.out.println(i);
i++;
}
```

Java Arrays

Arrays are used to store multiple values in a single variable, instead of declaring separate variables for each value.

To declare an array, define the variable type with square brackets:

String[] cars;

We have now declared a variable that holds an array of strings. To insert values to it, you can place the values in a comma-separated list, inside curly braces:

```
String[] cars = {"Volvo", "BMW", "Ford", "Mazda"};
```

To create an array of integers, you could write:

```
int[ ] myNum = {10,20,30,40};
```

Access the Elements of an Array

You can access an array element by referring to the index number.

This statement accesses the value of the first element in cars:

Example

```
String[] cars = {"Volvo", "BMW", "Ford", "Mazda"};
System.out.println(cars[0]);
// Outputs Volvo
```

Try it Yourself "

Note: Array indexes start with 0: [0] is the first element. [1] is the second element, etc.

Change an Array Element

To change the value of a specific element, refer to the index number:

Example

```
cars[0] = "Opel";
```

Example

```
String[] cars = {"Volvo", "BMW", "Ford", "Mazda"};
cars[0] = "Opel";
System.out.println(cars[0]);
// Now outputs Opel instead of Volvo
```

Try it Yourself

Array Length

To find out how many elements an array has, use the length property:

Example

```
String[] cars = {"Volvo", "BMW", "Ford", "Mazda"};
System.out.println(cars.length);
// Outputs 4
```

JAVA Classes, Overloading and Inheritance

Objectives: At the end of this lesson you shall be able to

- java OOPs Concept
- object Cloning
- types of Method.

Java Object Class

Java OOPs Concept

We will learn about the basics of OOPs. Object-Oriented Programming is a paradigm (Theory) that provides many concepts (ideas), such as inheritance, data binding, polymorphism, etc.

Simula is considered the first object-oriented programming language. The programming paradigm where everything is represented as an object is known as a truly object-oriented programming language.

Smalltalk is considered the first truly object-oriented programming language. The popular object-oriented languages are Java, C#, PHP, Python, C++, etc.

The main aim of object-oriented programming is to implement real-world entities, for example, object, classes, abstraction, inheritance, polymorphism, abstraction, encapsulation, etc.

OOPs (Object-Oriented Programming System)

Object means a real-world entity such as a pen, chair, table, computer, watch, etc. Object-Oriented Programming is a methodology or paradigm to design a program using classes and objects. It simplifies software development and maintenance by providing some concepts:

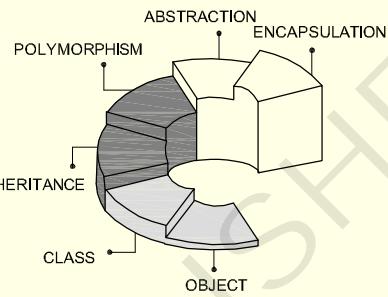
- Object
- Class
- Inheritance
- Polymorphism
- Abstraction
- Encapsulation

Apart from these concepts, there are some other terms which are used in Object-Oriented design:

- Coupling
- Cohesion
- Association
- Aggregation
- Composition

Java OOPs Concepts

Fig 1



CP20N1411711

Object

Any entity that has state and behavior is known as an object. For example, a chair, pen, table, keyboard, bike, etc. It can be physical or logical.

An Object can be defined as an instance of a class. An object contains an address and takes up some space in memory. Objects can communicate without knowing the details of each other's data or code. The only necessary thing is the type of message accepted and the type of response returned by the objects.

Example: A dog is an object because it has states like color, name, breed, etc. as well as behaviors like wagging the tail, barking, eating, etc.

Class

Collection of objects is called class. It is a logical entity.

A class can also be defined as a blueprint from which you can create an individual object. Class doesn't consume any space.

Inheritance

When one object acquires all the properties and behaviours of a parent object, it is known as inheritance. It provides code reusability. It is used to achieve runtime polymorphism.

Polymorphism

If one task is performed in different ways, it is known as polymorphism. For example: to convince the customer differently, to draw something, for example, shape, triangle, rectangle, etc.

In Java, we use method overloading and method overriding to achieve polymorphism.

Another example can be to speak something; for example, a cat speaks meow, dog barks woof, etc.

Abstraction

Hiding internal details and showing functionality is known as abstraction. For example, phone call, we don't know the internal processing.

In Java, we use abstract class and interface to achieve abstraction.

Encapsulation

Binding (or wrapping) code and data together into a single unit are known as encapsulation. For example, a capsule, it is wrapped with different medicines.

A java class is the example of encapsulation. Java bean is the fully encapsulated class because all the data members are private here.

Coupling

Coupling refers to the knowledge or information or dependency of another class. It arises when classes are aware of each other. If a class has the details information of another class, there is strong coupling. In Java, we use private, protected, and public modifiers to display the visibility level of a class, method, and field. You can use interfaces for the weaker coupling because there is no concrete implementation.

Cohesion

Cohesion refers to the level of a component which performs a single well-defined task. A single well-defined task is done by a highly cohesive method. The weakly cohesive method will split the task into separate parts. The `java.io` package is a highly cohesive package because it has I/O related classes and interface. However, the `java.util` package is a weakly cohesive package because it has unrelated classes and interfaces.

Association

Association represents the relationship between the objects. Here, one object can be associated with one object or many objects. There can be four types of association between the objects:

- One to One
- One to Many
- Many to One, and
- Many to Many

Let's understand the relationship with real-time examples. For example, One country can have one prime minister (one to one), and a prime minister can have many ministers (one to many). Also, many MP's can have one prime minister (many to one), and many ministers can have many departments (many to many).

Association can be unidirectional or bidirectional.

Aggregation

Aggregation is a way to achieve Association. Aggregation represents the relationship where one object contains other objects as a part of its state. It represents the weak relationship between objects. It is also termed as a has-

a relationship in Java. Like, inheritance represents the is-a relationship. It is another way to reuse objects.

Composition

The composition is also a way to achieve Association. The composition represents the relationship where one object contains other objects as a part of its state. There is a strong relationship between the containing object and the dependent object. It is the state where containing objects do not have an independent existence. If you delete the parent object, all the child objects will be deleted automatically.

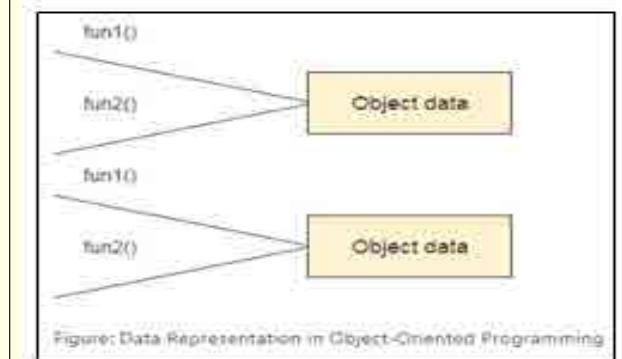
Advantage of OOPs over Procedure-oriented programming language

- 1 OOPs makes development and maintenance easier, whereas, in a procedure-oriented programming language, it is not easy to manage if code grows as project size increases.
- 2 OOPs provides data hiding, whereas, in a procedure-oriented programming language, global data can be accessed from anywhere.
- 3 OOPs provides the ability to simulate real-world events much more effectively. We can provide the solution of real-world problem if we are using the Object-Oriented Programming language.

GlobalData

Figure: Data Representation in Procedure-Oriented Programming (Fig 2)

Fig 2



ObjectData

What is the difference between an object-oriented programming language and object-based programming language?

Object-based programming language follows all the features of OOPs except Inheritance. JavaScript and VB Script are examples of object-based programming languages.

Do You Know?

- Can we overload the main method?
- A Java Constructor returns a value but what?
- Can we create a program without main method?
- What are the six ways to use this keyword?
- Why is multiple inheritance not supported in Java?
- Why use aggregation?
- Can we override the static method?
- What is the covariant return type?
- What are the three usages of Java super keyword?
- Why use instance initializer block?
- What is the usage of a blank final variable?
- What is a marker or tagged interface?
- What is run time polymorphism or dynamic method dispatch?
- What is the difference between static and dynamic binding?
- How downcasting is possible in Java?
- What is the purpose of a private constructor?
- What is object cloning?

What will we learn in OOPs Concepts?

- Advantage of OOPs
- Naming Convention
- Object and class
- Method overloading
- Constructor
- static keyword
- this keyword with six usage
- Inheritance
- Aggregation
- Method Overriding
- Covariant Return Type
- super keyword

- InstanceInitializerBlock
- final keyword
- Abstract class
- Interface
- Runtime Polymorphism
- Static and Dynamic Binding
- Downcasting with instanceof operator
- Package
- Access Modifiers
- Encapsulation

Object Cloning

Method in Java

In general, a method is a way to perform some task. Similarly, the method in Java is a collection of instructions that performs a specific task. It provides the reusability of code. We can also easily modify code using methods. In this section, we will learn what is a method in Java, types of methods, method declaration, and how to call a method in Java.

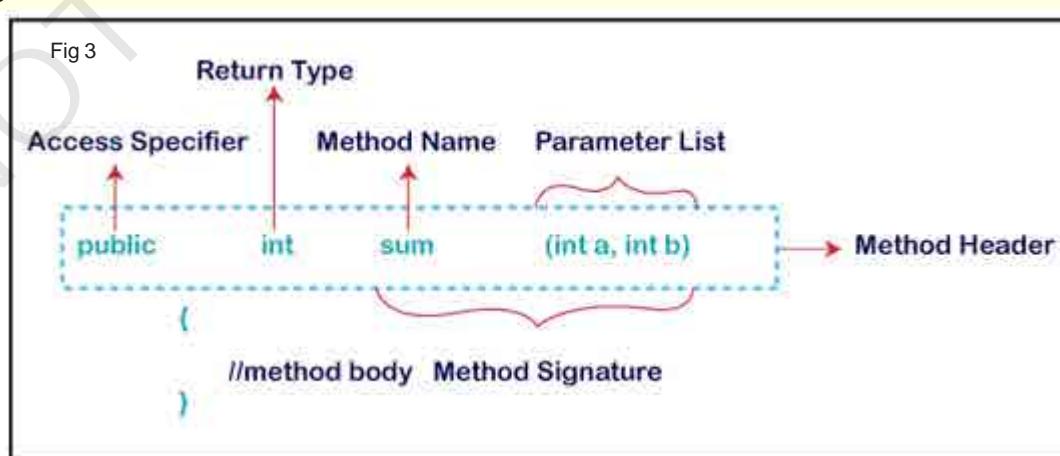
What is a method in Java?

A method is a block of code or collection of statements or a set of code grouped together to perform a certain task or operation. It is used to achieve the reusability of code. We write a method once and use it many times. We do not require to write code again and again. It also provides the easy modification and readability of code, just by adding or removing a chunk of code. The method is executed only when we call or invoke it.

The most important method in Java is the main() method. If you want to read more about the main() method, go through the link <https://www.javatpoint.com/java-main-method>.

Method Declaration (Fig 3)

The method declaration provides information about method attributes, such as visibility, return-type, name, and arguments. It has six components that are known as method header, as we have shown in the following figure.



Method Signature: Every method has a method signature. It is a part of the method declaration. It includes the method name and parameter list.

Access Specifier: Access specifier or modifier is the access type of the method. It specifies the visibility of the method. Java provides four types of access specifier:

- **Public:** The method is accessible by all classes when we use public specifier in our application.
- **Private:** When we use a private access specifier, the method is accessible only in the classes in which it is defined.
- **Protected:** When we use protected access specifier, the method is accessible within the same package or subclasses in a different package.
- **Default:** When we do not use any access specifier in the method declaration, Java uses default access specifier by default. It is visible only from the same package only.
- **Return Type:** Return type is a data type that the method returns. It may have a primitive data type, object, collection, void, etc. If the method does not return anything, we use void keyword.
- **Method Name:** It is a unique name that is used to define the name of a method. It must be corresponding to the functionality of the method. Suppose, if we are creating a method for subtraction of two numbers, the method name must be subtraction(). A method is invoked by its name.
- **Parameter List:** It is the list of parameters separated by a comma and enclosed in the pair of parentheses. It contains the data type and variable name. If the method has no parameter, left the parentheses blank.
- **Method Body:** It is a part of the method declaration. It contains all the actions to be performed. It is enclosed within the pair of curly braces.

Naming a Method

While defining a method, remember that the method name must be a verb and start with a lowercase letter. If the method name has more than two words, the first name must be a verb followed by adjective or noun. In the multi-word method name, the first letter of each word must be in uppercase except the first word. For example:

Single-word method name: sum(), area()

Multi-word method name: areaOfCircle(), stringComparison()

It is also possible that a method has the same name as another method name in the same class, it is known as method overloading.

Types of Method

There are two types of methods in Java:

- Predefined Method
- User-defined Method

Predefined Method

In Java, predefined methods are the method that is already defined in the Java class libraries is known as predefined methods. It is also known as the standard library method or built-in method. We can directly use these methods just by calling them in the program at any point. Some predefined methods are length(), equals(), compareTo(), sqrt(), etc. When we call any of the predefined methods in our program, a series of codes related to the corresponding method runs in the background that is already stored in the library.

Each and every predefined method is defined inside a class. Such as print() method is defined in the java.io.PrintStream class. It prints the statement that we write inside the method. For example, print("Java"), it prints Java on the console.

Let's see an example of the predefined method.

Demo.java

```
public class Demo  
{  
    public static void main(String[] args)  
    {  
        // using the max() method of Math class  
        System.out.print ("The maximum number is: "  
                         + Math.max(9,7));  
    }  
}
```

Output:

The maximum number is: 9

In the above example, we have used three predefined methods main(), print(), and max(). We have used these methods directly without declaration because they are predefined. The print() method is a method of PrintStream class that prints the result on the console. The max() method is a method of the Math class that returns the greater of two numbers.

max

```
public static int max(int a,  
                     int b)
```

Returns the greater of two int values.
same value.

Parameters:

a - an argument,
b - another argument.

Returns:

the larger of a and b.

We can also see the method signature of any predefined method by using the link <https://docs.oracle.com/>. When we go through the link and see the max () method signature, we find the following:

In the above method signature, we see that the method signature has access specifier public, non-access modifier static, return type int, method name max(), parameter list (int a, int b). In the above example, instead of defining the method, we have just invoked the method. This is the advantage of a predefined method. It makes programming less complicated. Similarly, we can also see the method signature of the print () method.

User-defined Method

The method written by the user or programmer is known as a user-defined method. These methods are modified according to the requirement.

How to Create a User-defined Method

Let's create a user-defined method that checks the number is even or odd. First, we will define the method.

```
//userdefinedmethod  
public static void findEvenOdd(int num)  
{  
    //method body if(num%2==0)  
    System.out.println(num+" is even"); else  
    System.out.println(num+" is odd");  
}
```

We have defined the above method named findEvenOdd(). It has a parameter num of type int. The method does not return any value that's why we have used void. The method body contains the steps to check the number is even or odd. If the number is even, it prints the number is even, else prints the number is odd.

How to Call or Invoke a User-defined Method

Once we have defined a method, it should be called. The calling of a method in a program is simple. When we call or invoke a user-defined method, the program control transfer to the called method.

```
import  
java.util.Scanner; public  
class EvenOdd  
{  
    public static void main(String args[])  
    {  
        //creating Scanner class object Scannersc  
        an=new Scanner(System.in);  
        System.out.print("Enter the number:");  
        //reading value from the  
        userintnum=scan.nextInt();
```

```
//method  
calling findEvenOdd  
(num);  
}
```

In the above code snippet, as soon as the compiler reaches at line findEvenOdd (num), the control transfers to the method and gives the output accordingly.

Let's combine both snippets of codes in a single program and execute it.

EvenOdd.java

```
import  
java.util.Scanner; public  
class EvenOdd  
{  
    public static void main(String args[])  
    {  
        //creating Scanner class object Scannersc  
        an=new Scanner(System.in);  
        System.out.print("Enter the number:");  
        //reading value from  
        userintnum=scan.nextInt();  
        //method  
        calling findEvenOdd  
(num);  
    }  
    //userdefinedmethod  
    public static void findEvenOdd(int num)  
    {  
        //method  
        body if(num%2  
        ==0)  
        System.out.println(num+" is  
        even"); else  
        System.out.println(num+" is odd");  
    }  
}
```

Output 1:

```
Enter the number:  
1212 is even
```

Output 2:

```
Enter the number: 99
```

99 isodd

Let's see another program that return a value to the calling method.

In the following program, we have defined a method named add() that sum up the two numbers. It has two parameters n1 and n2 of integer type. The values of n1 and n2 correspond to the value of a and b, respectively. Therefore, the method adds the value of a and b and store it in the variables and returns the sum.

Addition.java

```
public class Addition
{
    public static void main(String[] args)
    {
        int a =
19; int b
=5;
//method calling
int c = add(a, b); //a and b are actual
parameters System.out.println("The sum of
a and b is=" + c);
}
//user-defined method
public static int add(int n1, int n2) //n1 and n2 are formal
parameters
{
    int
s; s=n1+n
2;
    return; //returning the sum
}
}
```

Output:

The sum of a and b is=24

Static Method

A method that has static keyword is known as static method. In other words, a method that belongs to a class rather than an instance of a class is known as a static method. We can also create a static method by using the keyword static before the method name.

The main advantage of a static method is that we can call it without creating an object. It can access static datamembers and also change the value of it. It is used to create an instance method. It is invoked by using the class name. The best example of a static method is the main () method.

Example of static

methodDisplay.java

```
public class Display
{
    public static void main(String[] args)
    {
        show();
    }
    static void show()
    {
        System.out.println("It is an example of static method.");
    }
}
```

Output:

It is an example of a static method.

InstanceMethod

The method of the class is known as an instance method. It is a non-static method defined in the class. Before calling or invoking the instance method, it is necessary to create an object of its class. Let's see an example of an instance method.

Instance Method Example.java

```
public class InstanceMethodExample
{
    public static void main(String[] args)
    {
        //Creating an object of the class
        InstanceMethodExample obj = new InstanceMethodExample();
        //invoking
        instanceMethod();
        System.out.println("The sum is:" + obj.add(12, 13));
    }
    int s;
    //user-defined method because we have not used static
    keyword
    public int add(int a, int b)
    {
        s = a+b;
        //returning the sum
        return s;
    }
}
```

Output:

The sum is:25

There are two types of instance method:

- Accessor Method
- Mutator Method

Accessor Method: The method(s) that reads the instance variable(s) is known as the accessor method. We can easily identify it because the method is prefixed with the word get. It is also known as getters. It returns the value of the private field. It is used to get the value of the private field.

Example

```
public int getId()  
{  
    return id;  
}
```

Mutator Method: The method(s) read the instance variable(s) and also modify the values. We can easily identify it because the method is prefixed with the word set. It is also known as setters or modifiers. It does not return anything. It accepts a parameter of the same data type that depends on the field. It is used to set the value of the private field.

Example

```
public void setRoll(int roll)  
{  
    this.roll=roll;  
}
```

Example of accessor and mutator method Student.java

```
public class Student  
{  
    private int roll; private String name;  
    public int getRoll()//accessormethod  
    {  
        return roll;  
    }  
    public void setRoll(int roll)//mutatormethod  
    {  
        this.roll=roll;  
    }  
    public String getName()  
    {  
        return name;  
    }  
    public void setName(String name)  
    {  
        this.name=name;  
    }  
    public void display()  
    {  
        System.out.println("Roll no.:  
"+roll);System.out.println("Student name:"+name);  
    }  
}
```

Abstract Classes and Interfaces in JAVA

Objectives: At the end of this lesson you shall be able to

- method
- types of inheritance
- rules for Overriding.

AbstractMethod

The method that does not has method body is known as abstract method. In other words, without an implementation is known as abstract method. It always declares in the abstract class. It means the class itself must be abstract if it has abstract method. To create an abstract method, we use the keyword `abstract`.

Syntax

```
abstract void method_name();
```

Example of abstract
method Demo.java

```
abstract class Demo//abstractclass
{
    //abstract method declaration abstract void display();
}

public class MyClass extends Demo
{
    //method implementation void display()
    {
        System.out.println("Abstractmethod?");
    }

    public static void main(String args[])
    {
        //creating object of abstract
        classDemo obj= new MyClass();
        //invoking abstract method obj.display();
    }
}
```

Output:

Abstractmethod...

Factorymethod

It is a method that returns an object to the class to which it belongs. All static methods are factory methods. For example, `NumberFormat obj = NumberFormat.getNumberInstance();`

Inheritance

Inheritance in Java is a mechanism in which one object acquires all the properties and behaviours of a parent object. It is an important part of OOPs (Object Oriented programming system).

The idea behind inheritance in Java is that you can create new classes that are built upon existing classes. When you inherit from an existing class, you can reuse methods and fields of the parent class. Moreover, you can add new methods and fields in your current class also.

Inheritance represents the IS-A relationship which is also known as a parent-child relationship.

Why use inheritance in java

- For Method Overriding (so run time polymorphism can be achieved).
- For Code Reusability.

Terms used in Inheritance

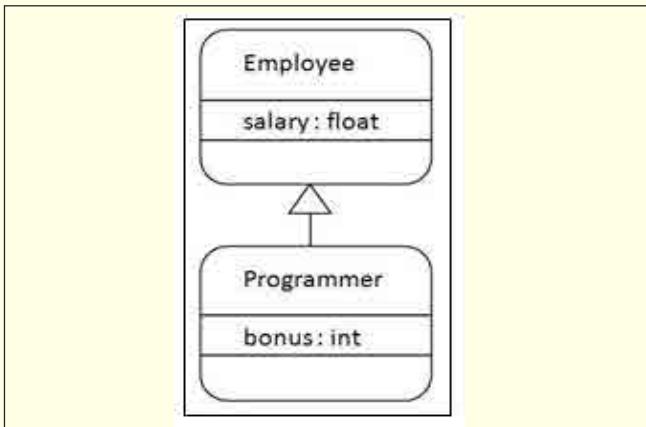
- **Class:** A class is a group of objects which have common properties. It is a template or blueprint from which objects are created.
- **Sub Class/Child Class:** Subclass is a class which inherits the other class. It is also called a derived class, extended class, or child class.
- **Super Class/Parent Class:** Superclass is the class from where a subclass inherits the features. It is also called a base class or a parent class.
- **Reusability:** As the name specifies, reusability is a mechanism which facilitates you to reuse the fields and methods of the existing class when you create a new class. You can use the same fields and methods already defined in the previous class.

The syntax of Java Inheritance

```
class Subclass-name extends Superclass-name
{
    //methods and fields
}
```

The `extends` keyword indicates that you are making a new class that derives from an existing class. The meaning of "extends" is to increase the functionality.

In the terminology of Java, a class which is inherited is called a parent or superclass, and the new class is called child or subclass.



Java Inheritance

Example

As displayed in the above figure, Programmer is the subclass and Employee is the superclass. The relationship between the two classes is Programmer IS-A Employee. It means that Programmer is a type of Employee.

```

class Employee {
}

float salary=40000;

}

class Programmer extends Employee{
int bonus=10000;
public static void main(String args[])
{
    Programmer p=new
    Programmer();System.out.println("Programmer salary
is:"+p.salary);System.out.print In ("Bonus of
Programmer is: "+p.bonus);
}
}

```

Output

```

Programmer salary
is:40000.0Bonus of programme
ris:10000

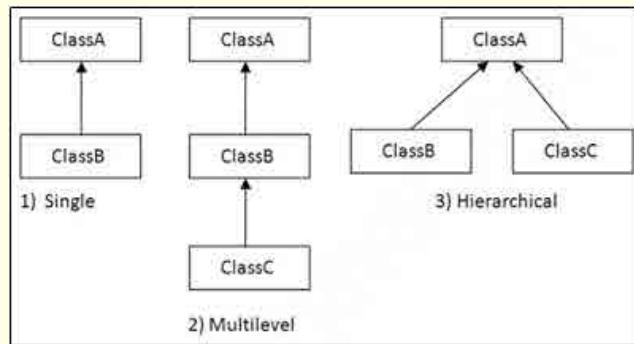
```

In the above example, Programmer object can access the field of own class as well as of Employee class i.e. code reusability.

Types of inheritance in java

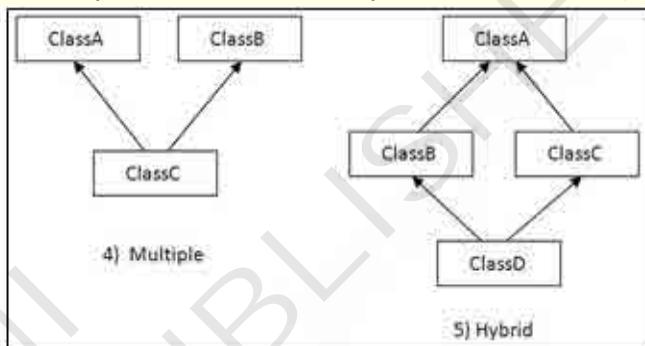
On the basis of class, there can be three types of inheritance in java: single, multilevel and hierarchical.

In java programming, multiple and hybrid inheritance is supported through interface only. We will learn about interfaces later.



Note: Multiple inheritance is not supported in Java through class.

When one class inherits multiple classes, it is known as multiple inheritance. For Example:



Single Inheritance Example

When a class inherits another class, it is known as a single inheritance. In the example given below, Dog class inherits the Animal class, so there is the single inheritance.

File: TestInheritance.java

```

class Animal{
void eat(){System.out.println("eating...");}
}

class Dog extends Animal{
void bark(){System.out.println("barking...");}
}

class TestInheritance{
public static void main(String
args[]){Dog d=new Dog();
d.bark();
d.eat();}
}

```

Output:

```

barking...
eating...

```

Multilevel Inheritance Example

When there is a chain of inheritance, it is known as multilevel inheritance. As you can see in the example given below, BabyDog class inherits the Dog class which again inherits the Animal class, so there is a multilevel inheritance.

File:TestInheritance2.java

```
class Animal{  
    void eat(){System.out.println("eating...");}  
}  
  
class Dog extends Animal{  
    void bark(){System.out.println("barking...");}  
}  
  
class BabyDog extends Dog{  
    void weep(){System.out.println("weeping...");}  
}  
  
class TestInheritance2{  
    public static void main(String  
    args[]){BabyDog d=new  
    BabyDog();d.weep();  
    d.bark();  
    d.eat();  
}}
```

Output: weeping... barking... eating...

Hierarchical Inheritance Example

When two or more classes inherit a single class, it is known as hierarchical inheritance. In the example given below, Dog and Cat classes inherit the Animal class, so there is hierarchical inheritance.

File:TestInheritance3.java

```
class Animal{  
    void eat(){System.out.println("eating...");}  
}  
  
class Dog extends Animal{  
    void bark(){System.out.println("barking...");}  
}  
  
class Cat extends Animal{  
    void meow(){System.out.println("meowing...");}  
}  
  
class TestInheritance3{  
    public static void main(String  
    args[]){Cat c=new Cat();  
    c.meow();  
    c.eat();  
}}
```

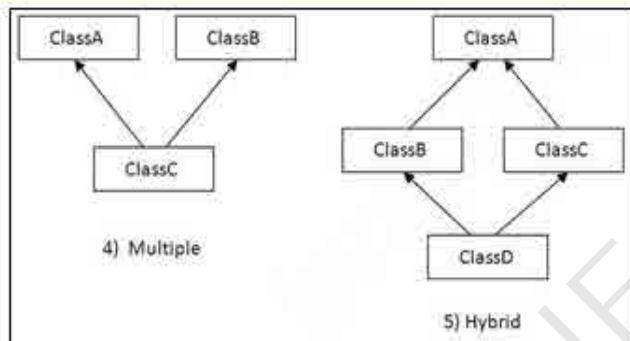
//c.bark()//C.T.Error

}

Output:

meowing...

eating...



Why multiple inheritance is not supported in java?

To reduce the complexity and simplify the language, multiple inheritance is not supported in java.

Consider a scenario where A, B, and C are three classes. The C class inherits A and B classes. If A and B classes have the same method and you call it from child class object, there will be ambiguity to call the method of A or B class.

Since compile-time errors are better than runtime errors, Java renders compile-time error if you inherit 2 classes. So whether you have same method or different, there will be compile time error.

```
class A{  
    void msg(){System.out.println("Hello");}  
}  
  
class B{  
    void msg(){System.out.println("Welcome");}  
}  
  
class C extends A,B{//suppose if it were  
public static void main(String args[]){Obj obj=new C();  
obj.msg();//Now which msg() method would be invoked?  
}}
```

CompileTime Error

Method Overriding in Java

If subclass (child class) has the same method as declared in the parent class, it is known as method overriding in Java.

In other words, if a subclass provides the specific implementation of the method that has been declared by one of its parent class, it is known as method overriding.

Usage of Java Method Overriding

- Method overriding is used to provide the specific implementation of a method which is already provided by its super class.

- Method overriding is used for runtime polymorphism

Rules for Java Method Overriding

- The method must have the same name as in the parent class.
- The method must have the same parameters as in the parent class.
- There must be an IS-A relationship (inheritance).

Understanding the problem without method overriding

Let's understand the problem that we may face in the program if we don't use method overriding.

//Java Program to demonstrate why we need method overriding

//Here, we are calling the method of parent class with child class object.

```
//Creating a parent class class Vehicle{
```

```
void run(){System.out.println("Vehicle is running");}
```

```
}
```

```
//Creating a child class class Bike extends Vehicle{
```

```
public static void main(String args[]){
```

```
//creating an instance of child class Bike obj = new Bike();
```

```
//calling the method with child class instance obj.run();
```

```
}
```

```
}
```

Output:

Vehicle is running

Problem is that I have to provide a specific implementation of run() method in subclass that is why we use method overriding.

Example of method overriding

In this example, we have defined the run method in the subclass as defined in the parent class but it has some specific implementation. The name and parameter

of the method are the same, and there is IS-A relationship between the classes, so there is method overriding.

//Java Program to illustrate the use of Java Method Overriding

Method Overriding in Java

//Creating a parent class.

```
class Vehicle{
```

```
//defining a method
```

```
void run(){System.out.println("Vehicle is running");}
```

```
}
```

```
//Creating a child class class Bike2 extends Vehicle{
```

```
//defining the same methods as in the parent class
```

```
void run(){System.out.println("Bike is running safely");}
```

```
public static void main(String
```

```
args[]){Bike2 obj = new
```

```
Bike2(); //creating
```

```
object obj.run(); //calling method
```

```
}
```

```
}
```

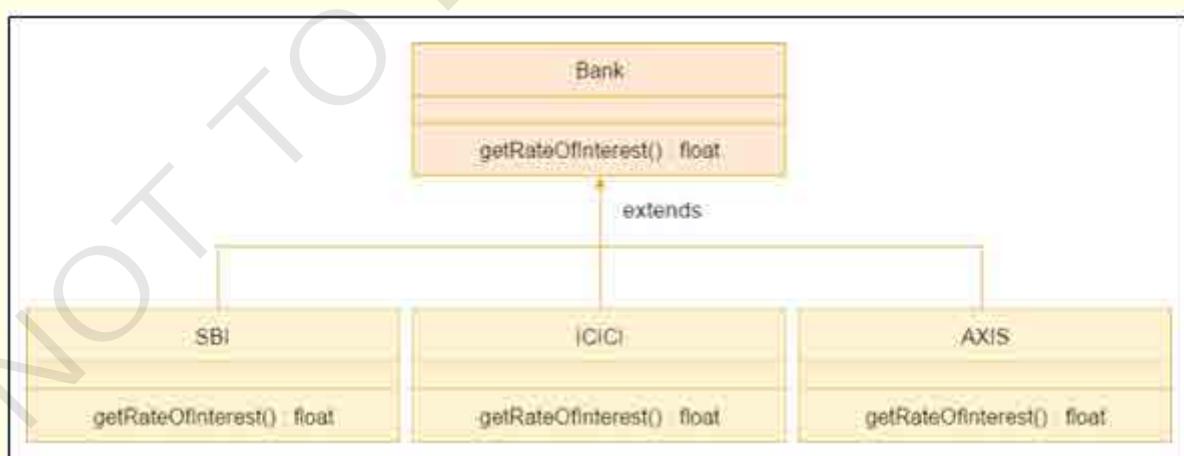
Output:

Bike is running safely

Area Example of Java Method Overriding

Consider a scenario where Bank is a class that provides functionality to get the rate of interest. However, the rate of interest varies according to banks. For example, SBI, ICICI and AXIS banks could provide 8%, 7%, and 9% rate of interest.

Note: Java method overriding is mostly used in Runtime Polymorphism which we will learn in next pages.



```

//Java Program to demonstrate the real scenario of Java
Method Overriding
//where three classes are overriding the method of a
parent class.
//Creating a parent
class.class Bank{
intgetRateOfInterest(){return0;}
}
//Creating child
classes.classSBI extends
Bank{
intgetRateOfInterest(){return8;}
}
classICIClextendsBank{
intgetRateOfInterest(){return7;}
}
classAXISextends Bank{
intgetRateOfInterest(){return9;}
}

//Test class to create objects and call the
methods
classTest2{
public static void main(String
args[])
{SBI s=newSBI();
ICICI i=new
ICICI();AXISA=newAXIS();
System.out.println("SBI Rate of Interest:
"+s.getRateOfInterest());System.out.println("ICICI Rate
of Interest:
"+i.get RateOfInterest()); System.out. println
("AXISRate of Interest:"+a.getRateOfInterest());
}
}

```

Output:

SBI Rate of Interest: 8ICICI Rate of Interest: 7AXISRate
ofInterest:9

Can we override static method?

No, a static method cannot be overridden. It can be proved by runtime polymorphism, so we will learn it later.

Why can we not override static method?

It is because the static method is bound with class whereas instance method is bound with an object. Static belongs to the class area, and an instance belongs to the heap area.

Can we override java main method?

No, because the main is a static method.

Difference between method Overloading and Method Overriding in java

Click me for the difference between method overloading and overriding to be taken

[More topics on Method Overriding \(Not for Beginners\)](#)

[Method Overriding with Access Modifier to be taken](#)

[Let's see the concept of method overriding with access modifier.](#) [Exception Handling with Method Overriding](#)

[Let's see the concept of method overriding with exception handling.](#)